

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

CARRERA: INGENIERÍA DE SISTEMAS

Tesis previa a la obtención del título de: INGENIEROS DE SISTEMAS

**TEMA:
ANÁLISIS, DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN
APLICATIVO MÓVIL PARA ADMINISTRACIÓN Y DISTRIBUCIÓN DE
NOTICIAS PARA LA EDITORIAL DON BOSCO**

**AUTORES:
EDWIN NORVEY CEBALLOS ANAGUANO
FRANCISCO RENÉ MIÑO MOSQUERA**

**DIRECTOR:
FRANKLIN EDMUNDO HURTADO LARREA**

Quito, febrero 2015

DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO DEL TRABAJO DE GRADO

Nosotros, autorizamos a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de grado y su reproducción sin fines de lucro.

Además declaramos que los conceptos y análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

Edwin Norvey Ceballos Anaguano
C.C. 171771879-3

Francisco René Miño Mosquera
C.C. 171625307-3

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Edwin Norvey Ceballos Anaguano y Francisco René Miño Mosquera, bajo mi dirección.

Ing. Franklin Hurtado
Director de Tesis

DEDICATORIA

“Conoceréis la verdad, y la verdad os hará libres”

Jesús de Nazaret

Dedico esta tesis esencialmente a mis padres que han sido el apoyo incondicional durante toda mi vida en todos los sentidos, a Dios que nos ha permitido la dicha de llegar hasta este punto y que bajo su amparo seguiremos hasta el fin del camino. Además agradezco a todas las personas que han sabido ser amigos, compañeros, camaradas y escuchas durante la realización de este trabajo y durante el proceso de aprendizaje, desarrollo emocional e intelectual de mi persona. Agradezco a todos los que han confiado en mí ya que su apoyo y palabras de aliento han sido un gran pilar para el logro de este éxito.

Edwin Norvey Ceballos Anaguano

El esfuerzo realizado y el logro conseguido en este largo proceso de trabajo y dedicación se lo ofrezco en primer lugar, a mis padres, que son mi familia y la base de mi vida. En segundo lugar, a aquellos amigos que supieron apoyarme cuando en verdad los necesitaba. Esta satisfacción y triunfo los comparto con ustedes, que al ser un motivo de inspiración y lucha, me permitieron finalizar esta etapa de mi vida.

Francisco René Miño Mosquera

AGRADECIMIENTOS

Al culminar la elaboración de esta Tesis queda la deuda de gratitud para muchas personas.

En primer lugar a nuestras familias, que en todo momento supieron acompañar en cada proyecto de nuestras vidas. Gracias por estar siempre a nuestro lado.

Gracias a aquellos amigos que estuvieron presentes a lo largo de nuestra formación, visible u ocultamente, aportando en diferente forma: material, espiritual o moral que muchas veces parecerían simples palabras de aliento.

Gracias a nuestro tutor, que supo guiarnos en la elaboración de este trabajo, de la manera más profesional y amigable posible.

Gracias a la Universidad Politécnica Salesiana por aportar con todo lo que necesitábamos en el desarrollo de la Tesis.

Y por encima de todo: gracias a DIOS por ser parte de nuestras vidas.

ÍNDICE

INTRODUCCIÓN.....	1
Antecedentes	2
Planteamiento del problema.....	3
Objetivos	3
Justificación.....	4
Alcance.....	5
CAPÍTULO 1.....	6
DESCRIPCIÓN DE LA EMPRESA	6
1.1 Generalidades.....	6
1.1.1 Visión	8
1.1.2 Misión	9
1.1.3 Actividad comercial	9
1.1.4 Productos y servicios.....	9
CAPÍTULO 2.....	11
MARCO TEÓRICO	11
2.1 Plataforma Android.....	11
2.1.1 Descripción histórica de Android.....	11
2.1.2 Arquitectura de Android	12
2.2 Métodos para análisis y desarrollo	14
2.2.1 Metodología Mobile-D.....	15
2.2.1.1 Principios de desarrollo de Mobile-D	15
2.2.1.2 Ciclo de vida de Mobile-D.....	15
2.2.1.3 Diagrama de bloques.....	17
2.2.1.4 Diagrama de navegabilidad.....	17
2.2.2 Lenguaje UML.....	17
2.2.2.1 Diagrama de casos de usos.....	17

2.2.2.2 Diagrama de clases.....	18
2.3 Tecnologías y herramientas para desarrollo.....	18
2.3.1 Lenguajes	18
2.3.1.1 Java.....	18
2.3.1.1.1 Definición de Java.....	18
2.3.1.1.2 Características de Java	19
2.3.1.1.3 Funcionamiento de Java.....	20
2.3.1.1.4 Ventajas de Java.....	21
2.3.1.2 XML.....	21
2.3.1.2.1 Definición de XML.....	21
2.3.1.2.2 Características de XML.....	21
2.3.1.2.3 Funcionamiento de XML	22
2.3.1.2.4 Ventajas de XML.....	23
2.3.2 Herramientas	23
2.3.2.1 Eclipse	23
2.3.2.1.1 Características	23
2.3.2.1.2 Ventajas.....	23
2.3.2.2 SDK Android	24
2.3.2.2.1 Características	24
2.3.2.2.2 Ventajas.....	25
2.3.2.3 Servicio web.....	25
2.3.2.3.1 Características	26
2.3.2.3.2 Funcionamiento.....	26
2.3.2.3.3 Ventajas.....	27
2.4 Definición de sistemas de administración de base de datos.....	27
2.4.1 SQLite	27
2.4.1.1 Características de SQLite.....	27

2.4.1.1.1 Tipos de datos	28
2.4.1.1.2 Ventajas de SQLite	28
2.5 Publicación de contenidos.....	29
2.5.1 Fuente de noticias sindicalizadas (RSS)	29
2.5.2 Canal de noticias electrónico	29
2.5.2.1 Políticas de publicación.....	30
2.5.2.2 Políticas de contenido	31
2.5.2.2.1 Información.....	32
2.5.2.2.2 Extensión.....	32
CAPÍTULO 3.....	33
ANÁLISIS Y DISEÑO.....	33
3.1 Obtención de requerimientos	33
3.1.1 Requerimientos no funcionales	34
3.1.1.1 Interfaces de usuario	36
3.1.1.2 Interfaces de software	41
3.1.1.3 Interfaces de comunicaciones	41
3.1.2 Requerimientos funcionales.....	41
3.1.2.1 Restricciones	46
3.2 Análisis.....	46
3.2.1 Propósito	46
3.2.2 Alcance.....	47
3.2.3 Visión general del documento.....	47
3.2.4 Descripción global	47
3.2.5 Perspectiva del producto	47
3.2.5.1 Ventajas.....	48
3.2.5.2 Beneficios.....	48
3.3 Diagramación	48

3.3.1 Diagramas de caso de uso	48
3.3.2 Diagrama de bloques.....	65
3.3.3 Diagrama de clases.....	66
3.3.4 Diagrama de navegabilidad.....	68
3.3.5 Diagrama de interfaz de usuario	69
CAPÍTULO 4.....	70
CONSTRUCCIÓN.....	70
4.1 Estándares de programación	70
4.1.1 Buenas prácticas para manejo de tablas	70
4.1.2 Buenas prácticas para manejo de instrucciones	71
4.1.3 Buenas prácticas para manejo de comentarios.....	71
4.1.4 Buena práctica para manejo de declaraciones.....	72
4.1.5 Regla de nomenclatura en variables y métodos	72
4.1.6 Regla de nomenclatura en clases, propiedades, componentes y controles	73
4.1.7 Regla de recuperación de errores	75
4.2 Arquitectura del sistema	76
4.2.1 Funcionamiento del servicio web EdiBosco	76
4.2.2 Análisis modelo vista controlador de aplicación móvil EdiBosco Noticias	80
4.3 Construcción de las soluciones de software	81
4.3.1 Construcción de la aplicación móvil	82
4.3.2 Construcción del servicio web	83
4.4 Construcción de interfaces de usuario.....	86
CAPÍTULO 5.....	88
PRUEBAS	88
5.1 Plan de pruebas	88
5.1.1 Ambiente de pruebas.....	91
5.1.2 Cronograma de trabajo.....	91

5.2 Pruebas funcionales.....	91
5.3 Pruebas no funcionales.....	101
5.4 Pruebas de rendimiento del servicio web.....	105
5.5 Resultado de pruebas.....	106
5.5.1 Resultado de pruebas funcionales	106
5.5.2 Resultado de pruebas no funcionales	108
5.5.3 Resultado de pruebas de rendimiento del servicio web	110
5.6 Publicación del aplicativo en la Web	112
5.7 Descarga e instalación del aplicativo por parte del usuario	112
Conclusiones	114
Recomendaciones	115
Lista de referencias	117
Anexos	120

ÍNDICE DE FIGURAS

Figura 1. Organigrama del Grupo Salesiano de Comunicación.....	7
Figura 2. Diagrama de unidades de servicio de Editorial Don Bosco.....	8
Figura 3. Estructura del Grupo Salesiano de Comunicación.	10
Figura 4. Arquitectura de la plataforma Android.	12
Figura 5. Orden y fases por las que pasa un proyecto Android hasta convertirse en aplicación.	13
Figura 6. Proceso completo de compilación de una aplicación Android.	14
Figura 7. Funcionamiento de aplicación móvil EdiBosco Noticias.	49
Figura 8. Diagrama de caso de uso Mostrar los grupos de noticias de las obras salesianas de forma ordenada.....	52
Figura 9. Diagrama de caso de uso Permitir al usuario conocer en que sección del aplicativo móvil se encuentra.....	54
Figura 10. Diagrama de caso de uso Brindar al usuario un conjunto de funciones que permitan visualizar, compartir, ordenar y refrescar las noticias.	55
Figura 11. Diagrama de caso de uso Mostrar al usuario los contenidos de las noticias de manera ordenada.....	56
Figura 12. Diagrama de caso de uso Brindar al usuario un listado de configuraciones aplicables a la información noticiosa.	58
Figura 13. Diagrama de caso de uso Brindar al usuario accesos directos para visualización e ingreso de nuevas noticias.....	59
Figura 14. Diagrama de caso de uso Brindar al usuario alertas o notificaciones de nuevas noticias.	61
Figura 15. Diagrama principal de caso de uso Administrar noticias.	62
Figura 16. Diagrama de bloques del sistema EdiBosco.	65
Figura 17. Diagrama de clases de aplicativo móvil EdiBosco Noticias.....	66
Figura 18. Diagrama de base de datos de aplicativo móvil EdiBosco Noticias.....	67
Figura 19. Diagrama de navegabilidad de aplicación móvil EdiBosco Noticias.	68
Figura 20. Diagrama de interfaz de usuario de aplicativo móvil EdiBosco Noticias.	69
Figura 21. Esquema de diseño del sistema EdiBosco Noticias.....	76
Figura 22. Arquitectura del servicio web EdiBosco.	79
Figura 23. Pantallas de aplicación móvil EdiBosco Noticias.....	86
Figura 24. Funciones principales de aplicación móvil EdiBosco Noticias.	87

Figura 25. Resultados obtenidos enfocados en el rendimiento del servicio web.....	110
Figura 26. Pantalla de registro de peticiones al servicio web EdiBosco durante 7 días.	111
Figura 27. Pantalla con los términos de licencia del entorno de desarrollo JDK.....	126
Figura 28. Pantalla de instalación del entorno de desarrollo JDK.	127
Figura 29. Pantalla de características que se desea instalar.	127
Figura 30. Pantalla de resultado de instalación del entorno de desarrollo JDK.	128
Figura 31. Sitio web del instalador del SDK de Android.....	129
Figura 32. Pantalla de instalación del SDK de Android.	129
Figura 33. Pantalla donde se selecciona el directorio de instalación del SDK de Android.	130
Figura 34. Pantalla con los paquetes de SDK de Android.	130
Figura 35. Pantalla de instalación de paquetes del SDK de Android.....	131
Figura 36. Pantalla con lista de dispositivos virtuales.	131
Figura 37. Pantalla de creación de dispositivos virtuales.....	132
Figura 38. Pantalla de ingreso de características del dispositivo virtual.....	133
Figura 39. Pantalla de comprobación de funcionamiento de dispositivo virtual.	133
Figura 40. Pantalla de ejecución del dispositivo virtual.....	134
Figura 41. Sitio web de descarga de Eclipse Juno 4.2.	135
Figura 42. Pantalla de instalación de plugin de Android.	136
Figura 43. Pantalla de asignación de nombre de repositorio que se va a agregar. ...	136
Figura 44. Pantalla de instalación de paquetes para desarrollo de aplicaciones Android.	137
Figura 45. Pantalla de nuevo proyecto de Android.	138
Figura 46. Pantalla de configuración de nuevo proyecto de Android.....	139
Figura 47. Pantalla de ejecución de aplicación móvil EdiBosco Noticias.	139
Figura 48. Pantalla con las condiciones de servicio y políticas de seguridad de Google.....	140
Figura 49. Pantalla con el acuerdo de distribución para desarrolladores de Google.	141
Figura 50. Pantalla de cuota de registro.	141
Figura 51. Pantalla donde se escoge la aplicación a subir al servidor.....	142
Figura 52. Pantalla de proceso de carga de aplicación al servidor de Google.	142
Figura 53. Pantalla donde se especifica si la aplicación es gratuita o de pago.	144

Figura 54. Ícono de aplicación móvil EdiBosco Noticias.....	145
Figura 55. Menú principal (sección Índice) de aplicación móvil EdiBosco Noticias.	145
Figura 56. Lista de noticias desplegadas en la sección Contenido.....	146
Figura 57. Detalle de una noticia seleccionada.	147
Figura 58. Pantalla de sección Configuración de aplicativo móvil EdiBosco Noticias.	148
Figura 59. Widget de aplicación móvil EdiBosco Noticias en el escritorio de Android.	149
Figura 60. Ícono de aplicación móvil EdiBosco Noticias en panel de notificaciones de dispositivo móvil.	149

ÍNDICE DE TABLAS

Tabla 1. Mostrar al usuario los grupos de noticias de las obras salesianas de forma ordenada.	42
Tabla 2. Permitir al usuario conocer en que sección del aplicativo móvil se encuentra.	43
Tabla 3. Brindar al usuario un conjunto de funciones que permitan visualizar, compartir, ordenar y refrescar las noticias.	43
Tabla 4. Mostrar al usuario los contenidos de las noticias de manera ordenada.	44
Tabla 5. Brindar al usuario un listado de configuraciones aplicables a la información noticiosa.	44
Tabla 6. Brindar al usuario accesos directos para visualización e ingreso de nuevas noticias.	45
Tabla 7. Brindar al usuario alertas o notificaciones de nuevas noticias.	45
Tabla 8. Actores principales que interactúan con el aplicativo móvil EdiBosco Noticias.	50
Tabla 9. Tareas de caso de uso principal.	51
Tabla 10. Controles y prefijos utilizados en tablas.	70
Tabla 11. Modelo vista controlador de aplicación móvil EdiBosco Noticias.	80
Tabla 12. Plan de pruebas.	89
Tabla 13. Cronograma de trabajo planificado para la ejecución de pruebas.	91
Tabla 14. Caso de prueba Mostrar al usuario los grupos de noticias de las obras salesianas de forma ordenada.	93
Tabla 15. Caso de prueba Permitir al usuario conocer en que sección del aplicativo móvil se encuentra.	94
Tabla 16. Caso de prueba Brindar al usuario un conjunto de funciones que permitan visualizar, compartir, ordenar y refrescar las noticias.	95
Tabla 17. Caso de prueba Mostrar al usuario los contenidos de las noticias de manera ordenada.	96
Tabla 18. Caso de prueba Brindar al usuario un listado de configuraciones aplicables a la información noticiosa.	96
Tabla 19. Caso de prueba Brindar al usuario accesos directos para visualización e ingreso de nuevas noticias.	97

Tabla 20. Caso de prueba Brindar al usuario alertas o notificaciones de nuevas noticias.	97
Tabla 21. Caso de prueba Administrar noticias.	98
Tabla 22. Caso de prueba Ingresar nueva noticia.....	98
Tabla 23. Caso de prueba Editar noticia.	99
Tabla 24. Caso de prueba Modificar noticia.	100
Tabla 25. Caso de prueba Eliminar noticia.	101
Tabla 26. Caso de prueba Rendimiento.	102
Tabla 27. Caso de prueba Seguridad.....	102
Tabla 28. Caso de prueba Fiabilidad.....	103
Tabla 29. Caso de prueba Disponibilidad.	103
Tabla 30. Caso de prueba Mantenibilidad.....	104
Tabla 31. Caso de prueba Portabilidad.....	104
Tabla 32. Caso de prueba Usabilidad.....	105
Tabla 33. Caso de prueba Rendimiento del servicio web.	106
Tabla 34. Resultados obtenidos enfocados en la funcionalidad del sistema con la participación del usuario.	107
Tabla 35. Resultados obtenidos enfocados en la funcionalidad del sistema con la participación del webmaster.....	108
Tabla 36. Resultados obtenidos enfocados en los requerimientos no funcionales...	109
Tabla 37. Resumen de evaluación de resultados.....	112
Tabla 38. Tabla Entrada.	123
Tabla 39. Detalles relación FK_ENTRADA_REFERENCE_FILTRO.....	123
Tabla 40. Tabla Noticia.....	124
Tabla 41. Detalles relación FK_FILTRO_REFERENCE_NOTICIA.	124
Tabla 42. Tabla Filtro.....	125

ÍNDICE DE ANEXOS

Anexo 1. Encuesta realizada a 20 personas para conocer el detalle de los servicios que ofrece el Grupo Salesiano de Comunicación.	120
Anexo 2. Diccionario de datos de EdiBosco Noticias.	123
Anexo 3. Instalación y configuración del entorno de desarrollo JDK.	126
Anexo 4. Instalación y configuración del SDK de Android.	129
Anexo 5. Instalación y configuración de Eclipse Juno.	135
Anexo 6. Instalación del plugin de Android para Eclipse.....	136
Anexo 7. Construcción del aplicativo EdiBosco Noticias.	138
Anexo 8. Creación de una cuenta desarrollador en Google para publicar la aplicación móvil en la tienda Google Play.	140
Anexo 9. Manual de usuario.	145

RESUMEN

Debido al constante crecimiento de la tecnología de la información y las comunicaciones en Ecuador, y en especial de los dispositivos móviles; las personas han visto la necesidad de encontrar nuevas formas de comunicación y de estar informadas de los productos y servicios que ofrece Editorial Don Bosco de una manera más ágil y rápida. El sistema de software aquí propuesto está enfocado a ayudar a suplir esa necesidad que tienen las personas en cuanto a estar informadas en cualquier momento a través de su dispositivo móvil de una forma agradable. Es por eso que el presente trabajo de titulación se centra en el análisis (recopilación y especificación de requerimientos), diseño (diagramación), construcción (desarrollo completo del aplicativo y los módulos necesarios que componen el mismo) e implementación (entrega del aplicativo a los usuarios a través de Internet) de una aplicación para la publicación de noticias en dispositivos de telefonía móvil con plataforma Android; es bueno mencionar que a través del sitio web informativo de la editorial se administrará toda la información que será presentada como noticia en el dispositivo móvil.

Adicionalmente el presente documento ayudará a quienes estén interesados en crear aplicaciones móviles para la plataforma Android, ya que se detalla los conceptos básicos para el aprendizaje de dicha plataforma.

ABSTRACT

Due to the constant growth of information technology and communications in Ecuador, especially for mobile devices; people have seen the need to find new ways to communicate and be informed of products and services Editorial Don Bosco offers a more agile and fast. The system proposed here is focused software to help meet that need that people have as to be informed at any time through your mobile device in a friendly manner. That is why this paper focuses on titration analysis (collection and specification requirements), design (layout), construction (complete development of the application and modules that compose it) and implementation (delivery of the application to users via Internet) of an application for publishing news on devices with Android mobile platform; is worth mentioning that through the informative website of the publishing any information that will be presented as news on the mobile device are given.

Additionally, this document will help those interested in creating mobile applications for the Android platform as the basics for learning the platform is detailed.

INTRODUCCIÓN

La proliferación y abaratamiento de los teléfonos móviles con capacidades para poder cargar diferentes tipos de software y con conexión a Internet han hecho que surjan, en los últimos tiempos, una gran variedad de aplicaciones que satisfacen muchas necesidades de los usuarios.

Considerando esta tendencia, es interesante y útil usar dispositivos móviles para informar a los usuarios de las diversas noticias que genera la Editorial Don Bosco. Dado que la empresa cuenta con un sitio web informativo que cumple este propósito, se busca establecer una nueva vía de comunicación con los usuarios, los cuales, según (Wayerless, 2012), la gran mayoría cuentan con dispositivos móviles con sistema operativo Android, sin embargo gran parte de éstos no tienen acceso al sitio web informativo, ocasionando demora al consultar las noticias.

Tomando en cuenta este inconveniente, se pretende construir un aplicativo móvil para la administración y distribución de noticias.

A través de esta aplicación móvil se puede presentar la información de una forma sencilla y rápida, de este modo los usuarios pueden recibir noticias actualizadas en tiempo real.

Se ha visto la necesidad de desarrollar la aplicación para dispositivos móviles utilizando tecnologías como XML y Java Script; y herramientas como Eclipse y SDK Android, lo cual permitió empaquetar todo dentro de una aplicación nativa para la visualización, obteniendo como resultado una aplicación más eficiente.

El presente proyecto de tesis se orienta bajo esta premisa, ofreciendo al usuario la visualización de noticias actuales que abarquen diferentes tópicos, las que son mostradas mediante texto e imágenes para una mayor comprensión de las mismas.

Se ha contemplado utilizar la metodología Mobile-D ya que está orientada a dispositivos móviles y permite el desarrollo ágil de software dado que dispone de un ciclo de vida muy rápido, el cual maneja cinco fases que son: Exploración: donde se

establece un plan de proyecto y los conceptos básicos, Inicialización: donde se preparan e identifican todos los recursos necesarios, Producción: donde se implementan todas las funcionalidades, Estabilización: donde se realizan acciones de integración para asegurar que el aplicativo completo funciona correctamente y Pruebas: donde se dispone de una versión estable y plenamente funcional del aplicativo.

La información que se muestra en este trabajo se encuentra distribuida en 5 capítulos, cuyos contenidos se describen a continuación.

El capítulo 1 comprende una breve descripción de la empresa.

El capítulo 2 (Exploración) comprende la descripción de la plataforma Android y de la metodología seleccionada para el modelamiento y posterior desarrollo del proyecto, finalizando con la descripción del IDE seleccionado.

El capítulo 3 (Inicialización) comprende la recopilación y especificación de requerimientos.

El capítulo 4 (Producción) comprende la construcción del aplicativo.

El capítulo 5 (Estabilización y Pruebas) comprende las pruebas de la aplicación realizadas en base a datos de prueba del aplicativo, posteriormente se evalúa la aplicación en un ambiente de prueba, con lo cual es posible determinar que el software cumple satisfactoriamente con la funcionalidad requerida.

Finalmente se plantean las conclusiones y recomendaciones al proyecto de titulación, así como, los anexos que permiten complementar la información plasmada en el presente proyecto.

Antecedentes

Editorial Don Bosco comienza sus actividades el 19 de Marzo de 1920, cuando un joven sacerdote, el padre Luis Natale Strazzieri, profesor del Colegio Cristóbal Colón de Guayaquil publicó su primer libro titulado *“Historia patria para el uso de los alumnos de segunda enseñanza”*, luego Librerías LNS, imprimió este libro en los talleres salesianos de tipografía y encuadernación de Quito. Con esta obra se inició la

prestigiosa colección de los textos escolares LNS tan conocidos en el Ecuador. Este acontecimiento marcó realmente el origen de Editorial Don Bosco en el Ecuador que tras varios años se ha convertido en una de las principales casas editoras que sigue publicando textos de enseñanza con el sello Luis Natale Strazzieri (LNS) para facilitar la tarea educativa de niños, adolescentes y jóvenes de escasos recursos (EdiBosco, 2013).

La empresa sigue en constante crecimiento y con el transcurrir del tiempo, ésta busca mejorar y establecer una vía de comunicación más eficiente con los usuarios que acceden a sus servicios, es por eso que el sitio web informativo de la editorial cumple con este propósito; sin embargo, no existe un acceso a sus noticias a través de un dispositivo móvil, lo que ha generado que la información no esté al alcance del usuario en cualquier momento.

Planteamiento del problema

Puesto que los canales de comunicación entre Editorial Don Bosco y sus clientes son muy limitados; no pueden abastecer a la mayoría de usuarios al momento de difundir sus noticias. Dado que dichos clientes no disponen de la facilidad, portabilidad y movilidad en este servicio, surge la necesidad de buscar una nueva vía de comunicación a través de un aplicativo móvil que administre y distribuya noticias referentes a las actividades de esta entidad. Aunque las consultas se las puede realizar a través de equipos de escritorio, este proceso demanda tiempo y eventualmente dinero y no brindan la comodidad que ofrece una aplicación móvil. Es por eso, según las nuevas tendencias del mercado que se observa la necesidad de crear un aplicativo móvil en la plataforma Android.

Objetivos

Los objetivos planteados en el proyecto son:

- Analizar, diseñar, construir e implementar un aplicativo móvil para la administración y distribución de noticias de la Editorial Don Bosco.
- Crear el aplicativo como una técnica de comunicación masiva destinada a difundir información con el fin de dar a conocer las actividades, productos y servicios que la editorial brinda.

- Brindar información exacta al usuario, ahorrándole tiempo y dinero.
- Mostrar al usuario la facilidad y comodidad al acceder a la información por medio del aplicativo.
- Realizar pruebas que verifiquen el funcionamiento óptimo del aplicativo móvil mediante el manejo y administración de datos reales.

Justificación

El presente trabajo es desarrollado con la finalidad de poder ofrecer a los usuarios de Editorial Don Bosco una nueva alternativa de poder realizar las consultas de las noticias.

Si bien es cierto, la editorial cuenta con un portal web informativo, al cual se puede ingresar solo si se tiene un ordenador conectado a Internet; dicho portal web consta de una aplicación web, manejada por el webmaster¹, el cual se encarga de administrar y distribuir las noticias; lo que se plantea en este trabajo de grado es diseñar un aplicativo móvil que recopile las noticias de la aplicación web para que el usuario las administre según sus necesidades y sea capaz de realizar algunas de las funciones del mencionado portal web con la diferencia de que la información pueda ser revisada mucho más rápidamente, en cualquier lugar y tiempo en el que el usuario lo solicite, ya que el principal propósito es de satisfacer los requerimientos de los usuarios de manera más rápida gracias al diseño de la aplicación.

En la actualidad muchas empresas han decidido trasladar sus aplicaciones a dispositivos móviles como una estrategia en el mercado, ya que, y tomando en cuenta la penetración de los dispositivos móviles en la población, buena parte de usuarios se identifican más con un aplicativo móvil que con un sitio web.

El software desarrollado podrá ser descargado en cualquier parte del mundo de forma gratuita, lo que permitirá fomentar la investigación a través del uso de aplicativos de dispositivos móviles y la riqueza técnica generada puede servir de base para en un futuro tener mejores resultados.

¹ Persona responsable del mantenimiento o programación de una aplicación web.

Actualmente, el avance de la tecnología en los dispositivos móviles permite tener acceso a Internet para poder enviar y recibir información de manera más rápida y confiable. Sin embargo, la editorial no aprovecha del todo las tecnologías existentes, lo cual puede originar una desventaja competitiva frente a otras editoriales.

Cabe mencionar que al momento se cuenta con la tecnología necesaria para desarrollar el aplicativo móvil, ya que existen varias herramientas como Eclipse y SDK Android que permiten diseñar la aplicación con mayor facilidad tomando en cuenta las nuevas tendencias tecnológicas.

Alcance

El proyecto pretende el desarrollo de un aplicativo móvil para la administración y distribución de noticias y tiene como principal desafío brindar a los usuarios la posibilidad de ver noticias actuales de carácter institucional, cultural y religioso en cualquier momento desde su dispositivo móvil. Este aplicativo mostrará noticias de la editorial y de sus principales proyectos como son Camino de Vida, encargado de responder con una propuesta innovadora, a la evangelización en las escuelas y colegios del país manejando las siguientes dimensiones: humanista, ético-moral e histórico-cultural y Esemtia, encargado del desarrollo de productos para Tecnologías de la Información y la Comunicación (TIC) en el ámbito educativo.

Cabe destacar que a pesar de todas las ventajas del aplicativo móvil, éste no reemplaza a ningún módulo de distribución de noticias usado por medio del sitio web informativo de la editorial, ni a ningún otro canal informativo de la institución.

El software está dirigido a todas aquellas personas que posean un dispositivo móvil, siempre y cuando éste cumpla con los requerimientos del sistema. Se asume también que toda persona que utilice este aplicativo deberá ser vidente y no tener impedimentos físicos que influyan directamente con el manejo del dispositivo.

CAPÍTULO 1

DESCRIPCIÓN DE LA EMPRESA

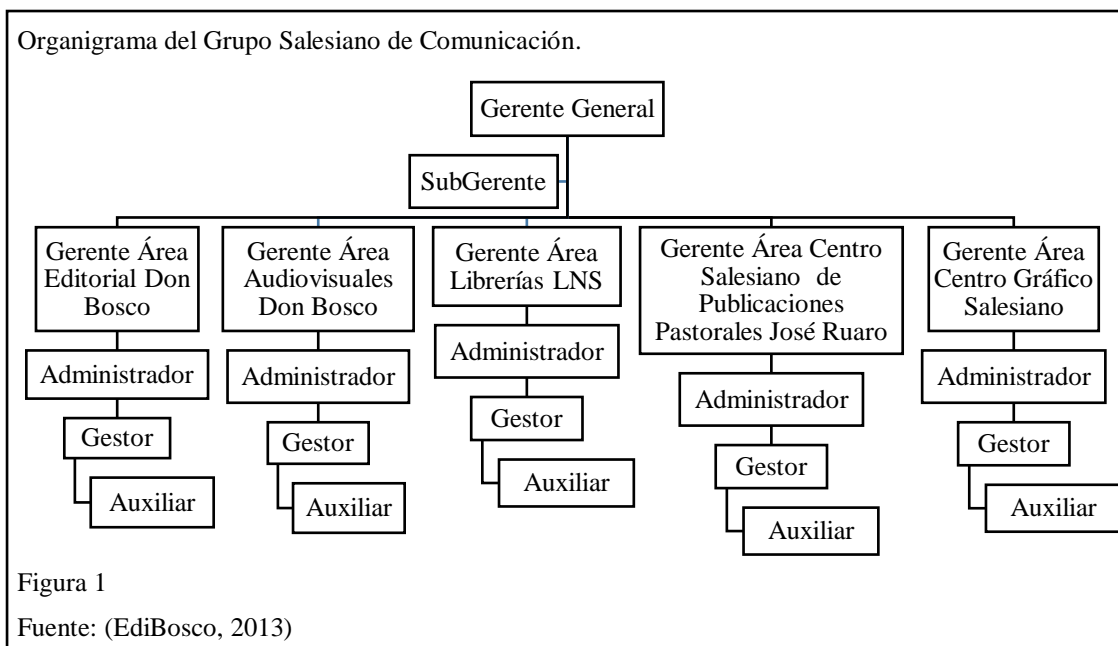
En este capítulo se describe, a breves rasgos, a la Editorial Don Bosco; su historia, misión, visión, actividad comercial y los productos y servicios que ofrece.

1.1 Generalidades

Desde sus inicios, el Grupo Salesiano de Comunicación ha diversificado paulatinamente sus actividades. Situaciones cambiantes del mercado y su propio crecimiento llevaron a esta empresa a crear nuevas divisiones, como se muestra en la figura 3, que pronto comenzaron a funcionar de forma independiente, las cuales se describen a continuación:

- **Editorial Don Bosco:** Es una empresa sin fines de lucro que publica y difunde toda clase de libros de texto y enseñanza, obras culturales y religiosas, y cualquier otro material didáctico necesario para orientar, asesorar y cooperar con la nación y con la iglesia en la tarea de educar a la juventud, especialmente de las clases populares.
- **Librerías LNS:** Es una empresa que se encarga de comercializar textos y material didáctico mediante sus canales de venta en las principales ciudades del país con una cobertura total, incluso registrando ventas a nivel latinoamericano.
- **Audiovisuales Don Bosco:** Es una productora de videos documentales educativos, pastorales y de desarrollo con más de 25 años de experiencia. Sus profesionales, equipamiento y experiencia permiten desarrollar productos de alta calidad.
- **Centro Salesiano de Publicaciones Pastorales José Ruaro:** Es una institución sin fines de lucro que se encarga de difundir el mensaje pastoral, el desarrollo de valores, buenos principios y la educación de jóvenes de escasos recursos a través de sus impresos.
- **Centro Gráfico Salesiano:** Es una empresa que se encarga de proveer soluciones impresas de calidad a precios competitivos de manera oportuna respetando el medio ambiente y siendo pioneros en responsabilidad social. (EdiBosco, 2013).

A continuación se muestra el organigrama del Grupo Salesiano de Comunicación, distribuido de la siguiente manera:



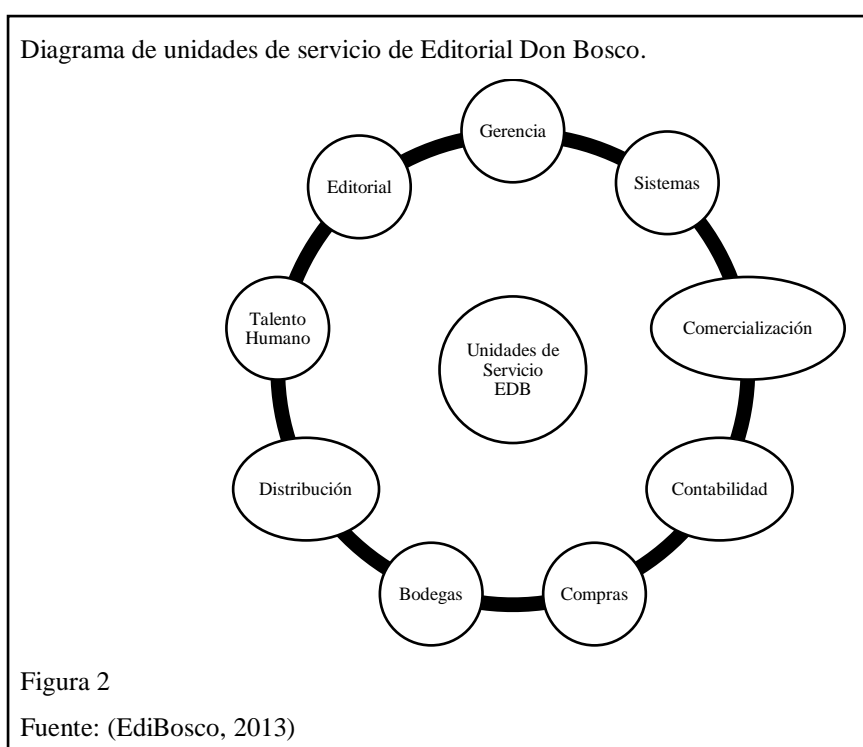
Con 93 años en el mercado, Editorial Don Bosco se ha transformado en una de las casas editoras más importantes del país por estar vinculada a la publicación de libros y revistas. La evaluación por parte de la editorial, la corrección de estilo y la revisión filológica, la diagramación, la composición, los acabados y la puesta a la venta y circulación son concretamente las fases del trabajo de esta entidad. Como se puede apreciar, su trabajo inicia desde el momento en el que el autor presenta su obra hasta que la misma se encuentra en el mercado. Además, Editorial Don Bosco organiza concursos para encontrar nuevos autores o premiar a los escritores más talentosos. Su actividad, por lo tanto, trasciende a la evaluación, gestión y publicación de libros. Cabe mencionar que en esta empresa laboran 60 personas, de las cuales 16 trabajan en el área administrativa, 20 en el área de producción y 24 en el área de comercialización. (EdiBosco, 2013).

La oficina principal de Editorial Don Bosco se encuentra ubicada en la ciudad de Cuenca, en la Avenida Vega Muñoz 10-68 y General Torres y su única sucursal se localiza en la ciudad de Quito en la Avenida 12 de Octubre N23-88 y Veintimilla. (EdiBosco, 2013).

Un hecho relevante se produjo en el año 2011 cuando Editorial Don Bosco publicó una serie de textos llamados “Camino de Vida”, los cuales van, desde 1ero de Básica

hasta 3ero de Bachillerato; cada texto está complementado con la guía para docentes, material de planificación y un recurso complementario para el aula de clase o la casa; destinados al crecimiento humano y espiritual de niños, jóvenes y maestros de diferentes escuelas y colegios salesianos del país. Este acontecimiento marcó el inicio de la expansión en el mercado nacional, ya que se produjo un incremento de un 40% en las ventas, mismo que es significativo y que hasta la fecha se mantiene relativamente estable. (EdiBosco, 2013).

Las unidades de servicio (departamentos) se encuentran distribuidos de la siguiente manera:



Entre sus principales unidades están, como lo indica la figura 2, el área editorial, encargada de la publicación de textos; el área de distribución, encargada de la entrega de material didáctico a los puntos de venta; y el área de comercialización, encargada de la venta de textos y material didáctico mediante todos los canales establecidos para este fin.

1.1.1 Visión

“Editorial Don Bosco es una empresa social, ética y transparente que pretende plasmar el enfoque del Sistema Preventivo aplicado por Don Bosco, el mismo que consiste en lograr cambios en las personas a partir de la vinculación con su entorno

ecológico, social y familiar, de modo que cada aprendizaje permita adquirir buenas relaciones con los demás, poner en práctica la libertad y la corresponsabilidad y desarrollar un pensamiento crítico” (EdiBosco, 2013).

1.1.2 Misión

“Editorial Don Bosco es una empresa consolidada y comprometida con los desafíos que impone la educación actual. Con productos diversificados y a la vanguardia del desarrollo educativo y tecnológico, con presencia en el mercado nacional e internacional” (EdiBosco, 2013).

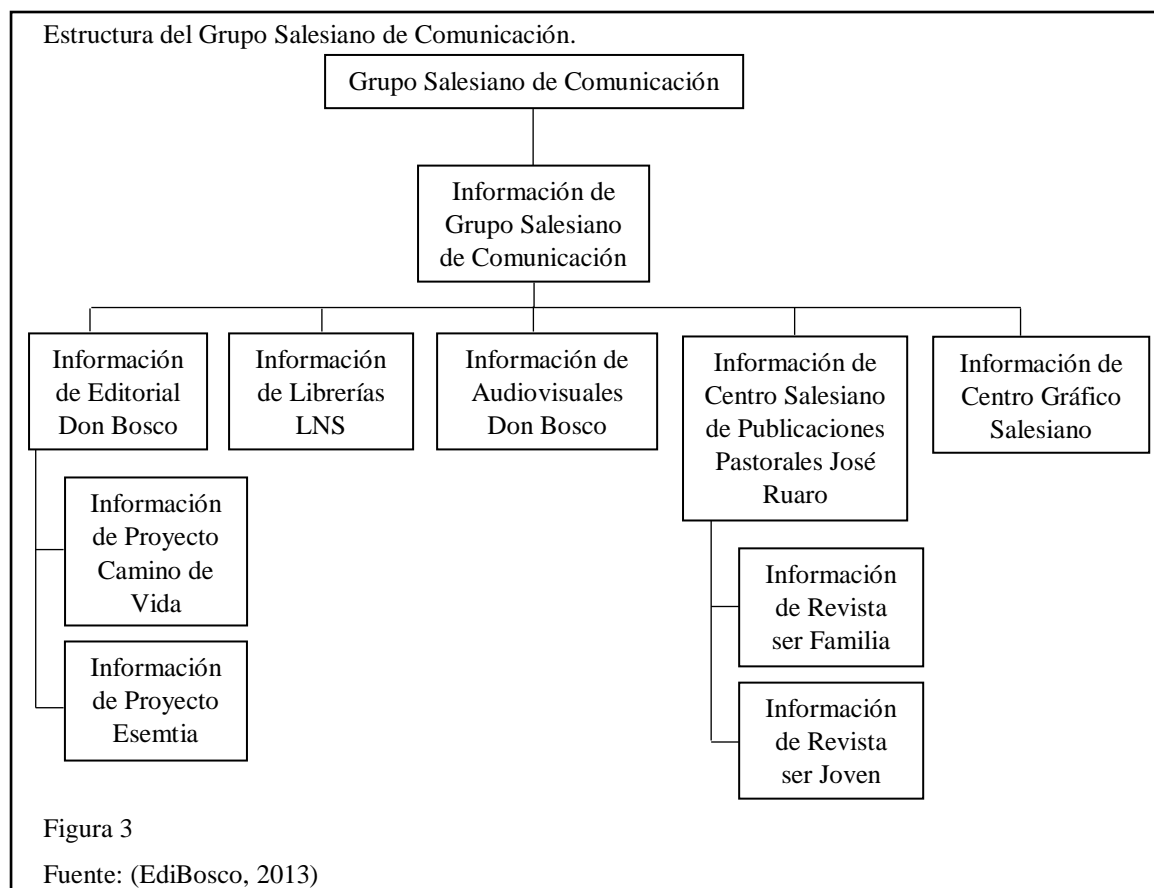
1.1.3 Actividad comercial

La editorial produce tres categorías de productos: textos escolares, material escolar multimedia y digital, y libros en general, que tienen por objetivo contribuir, de manera significativa, en la formación de honrados ciudadanos y buenos cristianos. De ahí que su línea editorial se orienta hacia tres frentes principales: la investigación y vanguardia educativa; la calidad e innovación pedagógica, científica y técnica; y el apego a los valores humanos y principios éticos que permiten forjar sociedades más justas y participativas (EdiBosco, 2013).

1.1.4 Productos y servicios

Los textos escolares y libros de Editorial Don Bosco persiguen un fin de servicio a la comunidad y son concebidos como una herramienta clave dentro del proceso de aprendizaje-enseñanza, sin distinción de calidad, según los grupos de mercado. Por esa razón, todos los textos y libros de Editorial Don Bosco apuntan a estándares de calidad elevados (de contenido, de diagramación, de impresión) e incorporan mensajes y lenguajes fácilmente comprensibles por padres de familia, docentes, niños, niñas, jóvenes y adultos de diversas realidades, lo que permite estrechar la brecha que ha existido tradicionalmente en la educación ecuatoriana.

La editorial cree en la formación continua, por lo que, conjuntamente con los textos escolares y libros publicados, ofrece planes y programas de capacitación, dirigidos a docentes, sobre temas pedagógicos, psicopedagógicos y de formación humana y relaciones interpersonales, entre otros de gran importancia para el desempeño efectivo de maestras y maestros en el aula y para el acercamiento a una educación de calidad (EdiBosco, 2013).



CAPÍTULO 2

MARCO TEÓRICO

En este capítulo se describe, de manera general, la plataforma Android, su historia y arquitectura, además se describe la metodología y los diagramas utilizados en este proceso de ingeniería, así como las tecnologías, herramientas y base de datos para el desarrollo de la aplicación móvil, de igual manera se indican las políticas de publicación y contenido de las noticias.

2.1 Plataforma Android

Android es una plataforma de código abierto, esto quiere decir, que cualquier desarrollador puede crear y desarrollar aplicaciones escritas con lenguaje Java u otros lenguajes y compilarlas (traducir un programa escrito en un lenguaje de programación a otro lenguaje de programación, generando un programa equivalente que un dispositivo será capaz de interpretar) a código nativo de ARM (conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción, son usadas generalmente en las bibliotecas de Android, API) (Andrade, 2010).

2.1.1 Descripción histórica de Android

Este sistema operativo fue desarrollado inicialmente por Android Inc., una firma comprada por Google (empresa especializada en productos y servicios relacionados con Internet, software, dispositivos electrónicos y otras tecnologías) en 2005. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio. La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (surface manager), un framework OpenCore, una base de datos relacional SQLite, una interfaz de programación de API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit (plataforma para aplicaciones que funciona como base para el navegador web Safari, Google Chrome, entre otros), un motor gráfico SGL, SSL (protocolos criptográficos que proporcionan comunicaciones seguras por

una red, comúnmente Internet) y una biblioteca estándar de C Bionic (Andrade, 2010).

2.1.2 Arquitectura de Android

Como se denota en la figura 4, la arquitectura de esta plataforma se basa en 4 niveles, los cuales son:

- **Aplicaciones:** Constituyen un conjunto de aplicaciones que corren en el dispositivo como base para otras o para el uso normal del smartphone. Estas son: cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Dentro de este nivel se desarrolló la aplicación.
- **Framework de aplicaciones:** Se encuentra separado por varios subsistemas y tienen un número de componentes específicos para cada módulo, por ejemplo el del administrador de teléfono, encargado de la gestión del dispositivo como hardware.
- **Bibliotecas de bajo nivel:** Escritas en C y C++, sirven para persistencia de datos, gestión de gráficos 3D, navegadores web embebidos. Entre estas librerías se encuentran elementos como SQLite, OpenGL, Webkit, etc.
- **Núcleo Linux (Kernel):** Sirve como una columna importante y fundamental de software la cual se encarga del sistema de gestión de drivers, seguridad, comunicaciones y algunas funciones esenciales del sistema (Petrazzini, 2012).

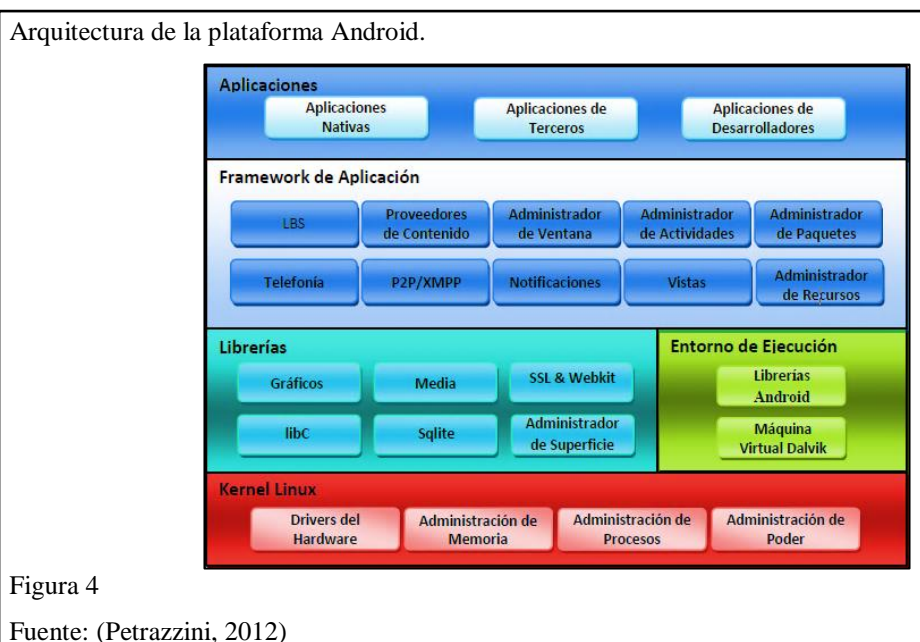
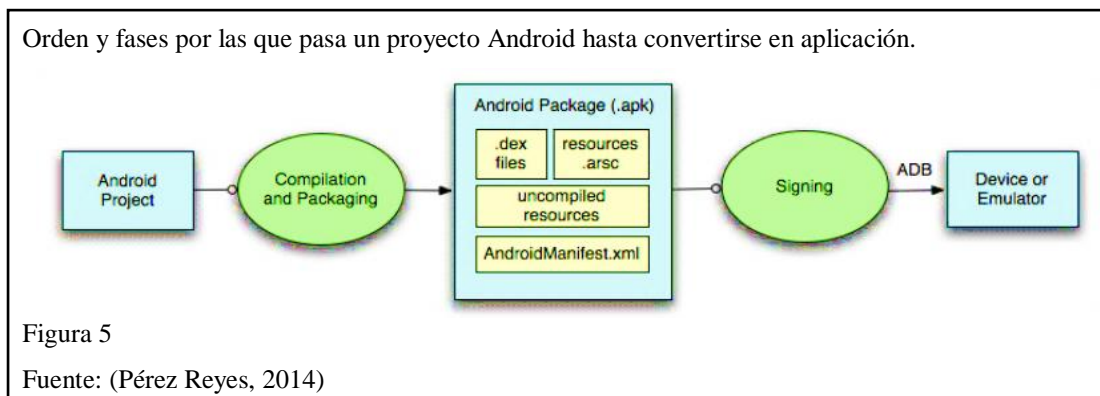


Figura 4

Fuente: (Petrazzini, 2012)

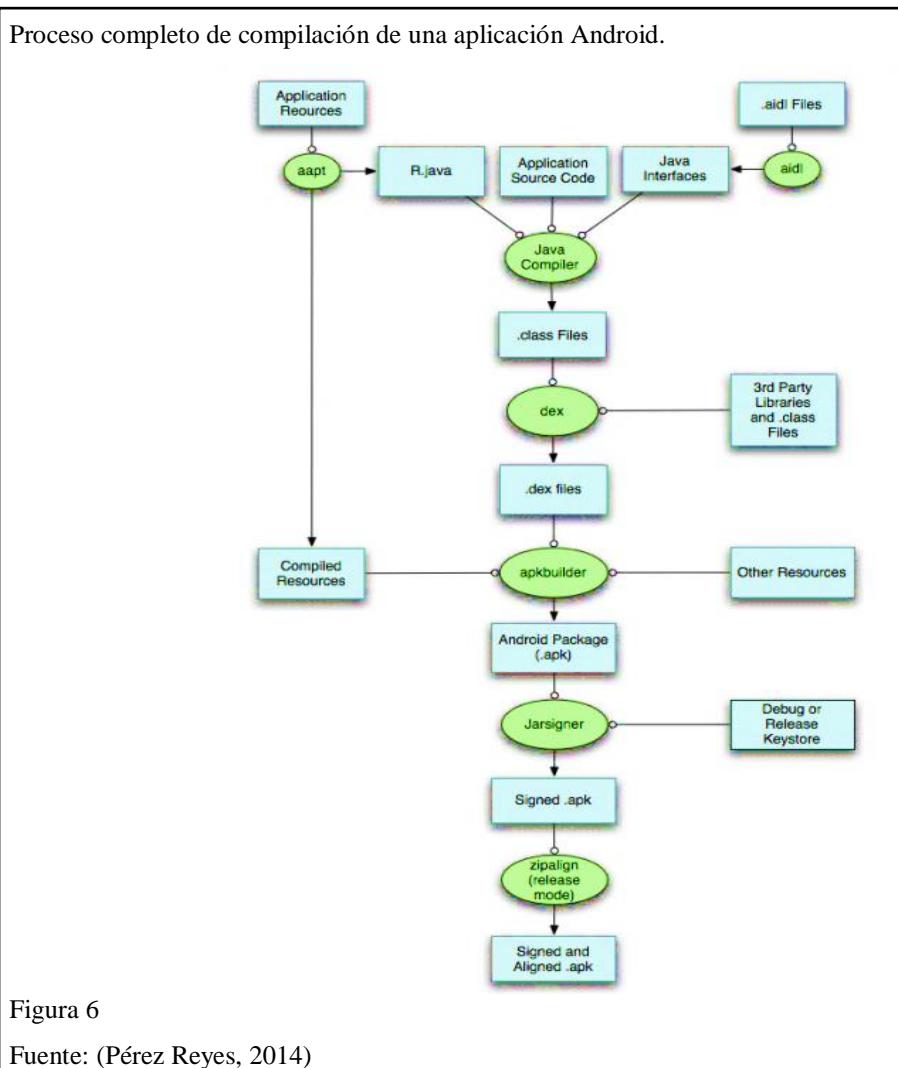
Una característica novedosa que tiene la plataforma Android es el uso de widgets o artilugios, los cuales son pequeñas aplicaciones interactivas que despliegan información directamente desde la pantalla principal. Aunque no son indispensables, entre sus objetivos está dar fácil acceso a funciones frecuentemente usadas y proveer de información visual al usuario. Los widgets usualmente vienen con una aplicación móvil; al descargar la aplicación también puede obtener el widget (Peña, 2012).

Durante la construcción de la aplicación, el presente proyecto se compiló y empaquetó en un fichero *.apk*, que contiene el binario de la aplicación. Este fichero contiene toda la información necesaria para poder lanzar la aplicación en un dispositivo móvil o emulador como lo muestra la figura 5 (Pérez Reyes, 2014).



El Android Asset Packaging Tool (aapt) recibió como entrada los recursos de la aplicación, como las imágenes, el *AndroidManifest.xml* y los documentos XML que definieron la interfaz de usuario y los compiló, produciendo el fichero *R.java* que permitió referenciar los recursos desde el código Java. El Android Asset Packaging Tool (aidl) convirtió cualquier interfaz *.aidl* en una interfaz de java. Todo el código Java incluyendo los ficheros *R.java* y *.aidl*, fueron compilados por el compilador de Java y produjeron los ficheros *.class*. La herramienta Dex convirtió los ficheros *.class* en Dalvik bytecode, un código intermedio para la máquina virtual Dalvik. Cualquier otra librería y fichero *.class* que esté incluido en el proyecto fue convertido en un fichero *.dex* para que pueda ser empaquetado en el instalador de la aplicación *.apk*. Una vez que se crea el fichero *.apk*, debió ser firmado con una llave de depuración o de lanzamiento antes de poder ser instalada en un dispositivo móvil. Para terminar, si la aplicación fue firmada en modo de lanzamiento, debió usar el Zipalign (herramienta de alineación de archivos que ofrece optimización importante de aplicaciones de Android (*.apk*) archivos) para alinear el fichero *.apk*, para reducir

el consumo de memoria cuando la aplicación se ejecuta en un dispositivo móvil como lo muestra la figura 6 (Pérez Reyes, 2014).



2.2 Métodos para análisis y desarrollo

Los métodos (conjunto de técnicas) para análisis y desarrollo de software hicieron su aparición alrededor de los años 60, con lo cual se buscaba la implementación de sistemas de información de una forma muy deliberada, estructurada y siguiendo una metodología (procedimientos), la cual requería que cada fase del ciclo de vida sea llevada a cabo de manera rígida y secuencial. En la actualidad los métodos para análisis y desarrollo de sistemas en especial de aplicaciones móviles, proveen un marco de trabajo, el cual es utilizado por las metodologías para estructurar, planear y controlar el proceso de desarrollo de un sistema informático, lo que incluye la predefinición de los entregables y artefactos que serán creados y completados por el equipo de desarrollo de una aplicación (Tudela, 2009).

2.2.1 Metodología Mobile-D

Mobile-D es una metodología de desarrollo ágil de software orientada para dispositivos móviles, se creó en un período de intenso crecimiento en el terreno de las aplicaciones móviles. Esta metodología apunta a la necesidad de disponer de un ciclo de desarrollo muy rápido, Mobile-D está pensado para grupos de no más de 10 desarrolladores colaborando en un mismo espacio físico. Si se trabaja con el ciclo de desarrollo propuesto, el proyecto debería finalizar con el lanzamiento del producto completamente funcional en poco tiempo (Lasso, 2013).

2.2.1.1 Principios de desarrollo de Mobile-D

Mobile-D se ha apoyado en muchas otras soluciones metodológicas bien conocidas y consolidadas, las cuales son: eXtreme Programming (XP), aporta con las prácticas de desarrollo, (Beck, 2000), Crystal Methodologies, aporta con el método de escalabilidad² y gestión de cambios, (Cockburn, 2004) y Rational Unified Process (RUP), aporta con la cobertura del ciclo de vida, (Kruchten, 1999).

2.2.1.2 Ciclo de vida de Mobile-D

El ciclo de esta metodología se divide en cinco fases: Exploración, Inicialización, Producción, Estabilización y Pruebas del aplicativo. En general, todas las fases (con la excepción de la primera) lleva poco tiempo de desarrollo. Se añadió tiempo para acciones adicionales en casos particulares.

La fase de Exploración, siendo ligeramente diferente del resto del proceso de producción, se dedica al establecimiento de un plan de proyecto y los conceptos básicos. Por lo tanto, se puede separar del ciclo principal de desarrollo.

Durante la fase de Inicialización, los desarrolladores preparan e identifican todos los recursos necesarios. Se preparan los planes para las siguientes fases y se establece el entorno técnico. La contribución al desarrollo ágil se centra fundamentalmente en esta fase, en la investigación de la línea arquitectónica. Esta acción se lleva a cabo durante el tiempo de planificación. Los desarrolladores analizan el conocimiento y los patrones arquitectónicos utilizados en la empresa y los relacionan con el proyecto actual. Se agregan las observaciones, se identifican similitudes y se extraen soluciones viables para su aplicación en el proyecto.

² Configuración para adaptarse a las circunstancias o requerimientos cambiantes.

En la fase de Producción se repite iterativamente hasta implementar todas las funcionalidades. Primero se planifica la iteración de trabajo en términos de requisitos y tareas a realizar. Las tareas se llevan a cabo en poco tiempo, desarrollando e integrando el código con el repositorio de datos existente. Luego se lleva a cabo la integración del aplicativo seguida de las pruebas de aceptación.

En la fase de Estabilización, se realizan las últimas acciones de integración para asegurar que el aplicativo completo funciona correctamente. En esta fase, los desarrolladores realizaron tareas similares a las que se desarrollan en la fase de producción, aunque en este caso todo el esfuerzo se dirige a la integración del aplicativo. Adicionalmente se puede considerar en esta fase la producción de documentación.

La fase de Pruebas tiene como meta la disponibilidad de una versión estable y plenamente funcional del aplicativo. El producto terminado e integrado se prueba con los requisitos del cliente y se eliminan todos los defectos encontrados (Lasso, 2013).

Las razones para ser la metodología seleccionada en el desarrollo del proyecto son las siguientes:

- Es una metodología ágil con ciclos de desarrollo cortos y para equipos pequeños.
- Está diseñado para el desarrollo de aplicaciones móviles.
- Facilidad para detectar y resolver tempranamente problemas técnicos.
- Baja densidad de defectos en las liberaciones de los productos.
- Se basa en el desarrollo basado en pruebas que es una de las mejores formas de asegurar la calidad.
- Se logra mejores diseños al basarse en el desarrollo basado en pruebas.
- Tiene un enfoque centrado en la satisfacción del usuario final, permitiendo mejorar el producto al realizar iteraciones cortas.
- Las tareas a realizar en cada fase están bien detalladas (Lasso, 2013).

Los principales artefactos que utiliza esta metodología son los diagramas de casos de uso y de clases, dado que son compatibles con el diseño de una aplicación móvil y se

ajustan a las fases que presenta la metodología Mobile-D; porque a través de estos diagramas se determinan los actores que intervienen en el software y las tareas que desempeñan cada uno de éstos, permitiendo que las interacciones en el proceso y las herramientas utilizadas sean más controlables y como resultado se debe tener una aplicación publicable y entregable al cliente, Mobile-D también contempla la utilización de diagramas de bloques y diagramas de navegabilidad.

2.2.1.3 Diagrama de bloques

El diagrama de bloques es la representación gráfica del funcionamiento interno de un sistema, que se hace mediante bloques y sus relaciones, y que, además, definen la organización de todo el proceso interno, sus entradas y sus salidas (Cuenca, 2007).

2.2.1.4 Diagrama de navegabilidad

Este diagrama es una herramienta útil para identificar la navegabilidad que existe entre las pantallas especificadas para la aplicación. Se construye a partir del direccionamiento que se describe para cada una de las pantallas que se definen previamente (Cuenca, 2007).

2.2.2 Lenguaje UML

UML es un lenguaje estándar de modelado de sistemas, que proporciona los planos de la arquitectura del sistema, puede utilizarse para visualizar, especificar, construir, y documentar procesos de negocios, funciones del sistema, lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables (Chacón, 2013).

2.2.2.1 Diagrama de casos de usos

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar. Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente (Cuenca, 2007).

2.2.2.2 Diagrama de clases

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia y de uso (Cuenca, 2007).

2.3 Tecnologías y herramientas para desarrollo

La correcta selección de las tecnologías y herramientas antes del proceso de desarrollo del aplicativo móvil constituye un factor muy importante, ya que de éstas va a depender en gran medida el rendimiento (tiempos de respuesta), portabilidad, fácil manejo y la terminación del sistema en el tiempo preestablecido con el cliente (Arco, 2012).

2.3.1 Lenguajes

Un lenguaje de programación es aquel que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente (Arco, 2012). Para este proyecto se utilizaron los siguientes lenguajes de programación:

2.3.1.1 Java

2.3.1.1.1 Definición de Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos, basado en clases y la primera plataforma informática creada por Sun Microsystems en 1995. Es la tecnología subyacente que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios. Java se ejecuta en más de 850 millones de ordenadores personales de todo el mundo y en miles de millones de dispositivos, como dispositivos móviles y aparatos de televisión (Cantero, 2011).

Fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como *WORA*, o "*write once, run anywhere*"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra (Cantero, 2011).

2.3.1.1.2 Características de Java

- **Lenguaje simple:** Java es más sencillo, ya que se han eliminado ciertas características, como los punteros. Debido a su semejanza con C y C++, resulta muy fácil aprender Java.
- **Orientado a objetos:** Java fue diseñado como un lenguaje orientado a objetos desde el principio. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos.
- **Distribuido:** Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.
- **Interpretado y compilado a la vez:** Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina, los bytecodes, semejantes a las instrucciones de ensamblador. Por otra parte, es interpretado, ya que los bytecodes se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (run-time).
- **Robusto:** Java fue diseñado para crear software altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución.
- **Seguro:** Dada la naturaleza distribuida de Java, donde las Applets se bajan desde cualquier punto de la red, la seguridad se impuso como una necesidad de vital importancia. Así que se implementaron barreras de seguridad en el lenguaje y en el sistema de ejecución en tiempo real.
- **Indiferente a la arquitectura:** Java está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, sobre arquitecturas distintas y con sistemas operativos diversos. El compilador de Java genera bytecodes: un formato intermedio indiferente a la arquitectura, diseñado para transportar el código eficientemente a múltiples plataformas de hardware y software.
- **Portable:** La indiferencia a la arquitectura representa sólo una parte de su portabilidad. Además, Java especifica los tamaños de sus tipos de datos

básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas.

- **Alto rendimiento**
 - **Multihebra:** Java soporta sincronización de múltiples hilos de ejecución (*multithreading*) a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas.
- **Dinámico:** El lenguaje Java y su sistema de ejecución en tiempo real son dinámicos en la fase de enlazado. Las clases sólo se enlazan a medida que son necesitadas. Se pueden enlazar nuevos módulos de código bajo demanda, procedente de fuentes muy variadas, incluso desde la red.
- **Produce Applets:** Las Applets son pequeños programas que aparecen embebidos en las páginas web, como aparecen los gráficos o el texto, pero con la capacidad de ejecutar acciones muy complejas, como animar imágenes, establecer conexiones de red, presentar menús y cuadros de diálogo para luego emprender acciones. (Cantero, 2011).

2.3.1.1.3 Funcionamiento de Java

El fichero de código fuente puede ser escrito mediante cualquier editor ASCII convencional o, para mayor comodidad, con el editor suministrado con el paquete del lenguaje Java. Una vez creado el fichero *.java* con el código del programa, se compila, generándose un fichero intermedio con los bytecodes, de extensión *.class*. Una vez generado el fichero *.class*, éste ya puede ser interpretado en cualquier máquina virtual de Java. Tratándose de Applets, este fichero se descargará típicamente desde la red. Para indicárselo al navegador se utilizan unas etiquetas especiales del lenguaje HTML.

```
<APPLET attributes>  
applet_parameters  
alternate_content  
</APPLET>
```

Esta página web en la que se encuentra embebida la Applet se baja desde la red o bien se carga desde el disco duro. En cualquier caso, para ver la Applet funcionando hace falta un navegador capaz de ejecutar Java, como los muy conocidos Mozilla y Microsoft Internet Explorer. El navegador carga las clases especificadas en la

etiqueta *<APPLET>* dinámicamente, a medida que van siendo necesitadas y se les pasa al código fuente (a los bytecodes) el verificador de código de bytes. Ahora será ya la máquina virtual de Java la que vaya interpretando los bytecodes y generando las instrucciones para su propia arquitectura (Cantero, 2011).

2.3.1.1.4 Ventajas de Java

- Java, un lenguaje de programación orientado a objetos es, sin duda, uno de los idiomas simple que es fácil de usar y es mucho más fácil de aprender, en comparación con otros idiomas.
- Para los desarrolladores de Java, Java lo hace muy conveniente y fácil de escribir programas de la red, que es tan sencillo como enviar y recibir datos hacia y desde un archivo.
- Otro aspecto que hace de Java en una plataforma ideal para el desarrollo de aplicaciones java es el factor de multiproceso. Esta característica permite que el programa multi-hilo para realizar una serie de tareas de forma simultánea (Cantero, 2011).

2.3.1.2 XML

2.3.1.2.1 Definición de XML

Es un lenguaje de marcas que ofrece un formato para la descripción de datos estructurados, es decir, XML es un metalenguaje, dado que con él se puede definir un propio lenguaje de presentación y, a diferencia del HTML, que se centra en la representación de la información, XML se centra en la información en sí misma. La particularidad más importante del XML es que no posee etiquetas prefijadas con anterioridad, ya que es el propio diseñador quien las crea a su antojo, dependiendo del contenido del documento (Cantero, 2011).

2.3.1.2.2 Características de XML

- Es una arquitectura más abierta y extensible. No se necesita versiones para que puedan funcionar en futuros navegadores
- Mayor consistencia, homogeneidad y amplitud de los identificadores descriptivos del documento con XML (Resource Description FrameWork RDF), en comparación a los atributos de la etiqueta del HTML.

- Integración de los datos de las fuentes más dispares. Se podrá hacer el intercambio de documentos entre las aplicaciones tanto en el propio PC como en una red local o extensa.
- Datos compuestos de múltiples aplicaciones. La extensibilidad y flexibilidad de este lenguaje permitirá agrupar una variedad amplia de aplicaciones, desde páginas web hasta bases de datos.
- Gestión y manipulación de los datos desde el propio cliente web.
- Los motores de búsqueda devolverán respuestas más adecuadas y precisas, ya que la codificación del contenido web en XML consigue que la estructura de la información resulte más accesible.
- Se permitirá un comportamiento más estable y actualizable de las aplicaciones web, incluyendo enlaces bidireccionales y almacenados de forma externa (El famoso epígrafe “404 file not found” desaparecerá).
- El concepto de “hipertexto” se desarrollará ampliamente, permitirá la denominación independiente de la ubicación, enlaces bidireccionales, enlaces que pueden especificarse y gestionarse desde fuera del documento, hiperenlaces múltiples, enlaces agrupados, atributos para los enlaces, etc.
- Exportabilidad a otros formatos de publicación (papel, web, cd-rom, etc.). El documento maestro de la edición electrónica podría ser un documento XML que se integraría en el formato deseado de manera directa (Cantero, 2011).

2.3.1.2.3 Funcionamiento de XML

- Es un estándar abierto, esto quiere decir que la gran mayoría de software de escritorio de sistema operativo y aplicaciones móviles, permiten la compatibilidad con XML esto lo hace muy potente a la hora de permitir la comunicación entre distintas plataformas de software y hardware.
- Simplicidad de sintaxis, esto significa que es muy fácil de escribir código en XML y la representación de los datos es casi entendible por cualquier ser humano.
- Independencia del protocolo de transporte, el hecho de que XML es un lenguaje de marcado de texto, no necesita de ningún protocolo de transporte especial, solo necesita de un protocolo que pueda transferir texto o documentos simples (Cantero, 2011).

2.3.1.2.4 Ventajas de XML

- Fácilmente procesable tanto por humanos como por software.
- Separa radicalmente la información o el contenido de su presentación o formato.
- Diseñado para ser utilizado en cualquier lenguaje o alfabeto.
- Su análisis sintáctico es fácil debido a las estrictas reglas que rigen la composición de un documento.
- Estructura jerárquica.
- El número de marcas es ilimitado (Cantero, 2011).

2.3.2 Herramientas

Las herramientas de desarrollo de software son programas informáticos que usa un programador para crear, depurar, gestionar o mantener un programa (Lasso, 2013).

Para este proyecto se utilizaron las siguientes herramientas de software:

2.3.2.1 Eclipse

Eclipse es un IDE multiplataforma para crear aplicaciones clientes de cualquier tipo. La primera y más importante aplicación que ha sido realizada con este entorno es el afamado IDE Java llamado Development Toolkit (JDT) y el compilador incluido en Eclipse (Tudela, 2009).

2.3.2.1.1 Características

- Dispone de un editor de texto con resaltado de sintaxis donde se puede ver el contenido del fichero en el que se está trabajando.
- Contiene una lista de tareas y otros módulos similares.
- La compilación es en tiempo real.
- Tiene pruebas unitarias con JUnit.
- Integración con Ant (herramienta usada en programación para la realización de tareas mecánicas y repetitivas, normalmente durante la fase de compilación y construcción), asistentes (wizards) para creación de proyectos, clases, tests, etc., y refactorización (Tudela, 2009).

2.3.2.1.2 Ventajas

- El entorno de desarrollo integrado (IDE) de Eclipse emplea módulos (en inglés plugin) para proporcionar toda su funcionalidad al frente de la

plataforma del cliente, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no.

- Este mecanismo de módulos es una plataforma ligera para componentes de software. Adicionalmente a permitirle a Eclipse extenderse usando otros lenguajes de programación como son C/C++ y Python, permite a Eclipse trabajar con lenguajes para procesamiento de texto como LaTeX, aplicaciones en red como Telnet y sistema de gestión de base de datos.
- La arquitectura plugin permite escribir cualquier extensión deseada en el ambiente, como sería gestión de la configuración. Se provee soporte para Java y CVS en el SDK de Eclipse. Y no tiene por qué ser usado únicamente para soportar otros lenguajes de programación (Tudela, 2009).

2.3.2.2 SDK Android

SDK Android es una plataforma para desarrollar aplicaciones y juegos para el sistema operativo de smartphones Android. El kit de desarrollo Android SDK incorpora todas las librerías y atributos necesarios para desarrollar, crear y probar aplicaciones para dispositivos Android, como también facilita probar aquellas aplicaciones que se están desarrollando en otro sistema (Andrade, 2010).

Para que SDK Android pueda funcionar correctamente es necesario instalar también una plataforma de desarrollo. Se recomienda el uso de Eclipse porque es la plataforma que oficialmente soporta SDK Android.

SDK Android incluye todo lo necesario para empezar a desarrollar aplicaciones como plugins, APIs, memoria y creación de perfiles de rendimiento, herramientas para la depuración y las necesarias para utilizar Java como lenguaje de programación. Además, es compatible con los formatos multimedia más comunes de audio, vídeo e imagen, y permite la reutilización y la sustitución de componentes (Andrade, 2010).

2.3.2.2.1 Características

- El Software Development Kit (SDK) de Android, incluye un conjunto de herramientas de desarrollo. Comprende un depurador de código, biblioteca, un simulador de teléfono basado en un emulador de procesadores basado en la traducción dinámica de binarios (QEMU), documentación, ejemplos de

código y tutoriales. Las plataformas de desarrollo soportadas incluyen Linux, Max OS X 10.4.9 o posterior, y Windows XP o posterior. La plataforma integral de desarrollo Integrated Development Environment (IDE) soportada oficialmente es Eclipse junto con el complemento Android Development Tools plugin (ADT), aunque también puede utilizarse un editor de texto para escribir ficheros Java y XML y utilizar comandos en un terminal para crear y depurar aplicaciones. Además, pueden controlarse dispositivos Android que estén conectados.

- Las actualizaciones del SDK están coordinadas con el desarrollo general de Android. El SDK soporta también versiones antiguas de Android, por si los programadores necesitan instalar aplicaciones en dispositivos ya obsoletos o más antiguos. Las herramientas de desarrollo son componentes descargables, de modo que una vez instalada la última versión, pueden instalarse versiones anteriores y hacer pruebas de compatibilidad. Una aplicación Android está compuesta por un conjunto de ficheros empaquetados en formato *.apk* y almacenada en el directorio */data/app* del sistema operativo Android. Un paquete APK incluye ficheros *.dex* (ejecutables Dalvik, un código intermedio compilado), recursos. (Andrade, 2010).

2.3.2.2.2 Ventajas

- SDK Android se encuentra disponible y al alcance para programadores que quieren desarrollar aplicaciones móviles y además proporciona todas las herramientas necesarias para la creación de las mismas por medio del lenguaje de programación Java.
- Las aplicaciones se elaboran y se desarrollan en SDK Android, luego se instalan en el dispositivo móvil configurado con este software. De esta manera el desarrollo de aplicaciones Android es completamente personalizado para las necesidades y requerimientos de los diferentes usuarios (Andrade, 2010).

2.3.2.3 Servicio web

Es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar

los servicios web para intercambiar datos en redes de ordenadores como Internet (Lavoe, 2009).

2.3.2.3.1 Características

- Basado en XML en toda su estructura.
- Basado en mensajes.
- Independiente del lenguaje de programación
- Puede ser localizado dinámicamente.
- Puede ser ensamblado o agregado dinámicamente.
- Accesible a través de Internet.
- Débilmente acoplados.
- Basado en estándares de la industria (Lavoe, 2009).

2.3.2.3.2 Funcionamiento

El cliente establece un diálogo coherente con el servicio web cliente (WSC), alojado en el servidor de aplicaciones, mediante el envío de una petición haciendo uso de un contrato llamado lenguaje de descripción de servicios web, este lenguaje está basado en el lenguaje XML el cual permite describir la interfaz pública de los servicios web (WSDL), para luego recibir la respuesta del servicio web servidor (WSS), el cual es el encargado de ejecutar los procesos y enviar respuestas al WSC. El WSDL es un archivo generado en una codificación XML el cual está basado en el protocolo Simple Object Access Protocol (SOAP). El servicio web (WS) no es como cuando un usuario navega en las páginas web, sino que el servicio web al recibir peticiones a través de mensajes SOAP desde otras aplicaciones, éste realiza el proceso solicitado y devuelve un mensaje SOAP de respuesta el cual puede ser utilizado por la aplicación cliente.

Es recomendable el uso del servicio web cuando hay necesidad de que un cliente autorizado acceda remotamente a la información y generar procesos, interactuando con el sistema que una empresa haya compartido para que pueda ser accedido en todo momento desde Internet. También, es recomendable el servicio web cuando se requiere crear una red virtual entre empresas, que tengan la necesidad de intercambiar información a través de Internet en un entorno seguro y protegido.

No es recomendable el uso del servicio web en sistemas donde no es requerido un entorno distribuido, como puede ser el compartir información a todos los usuarios sin restricciones en el cual no se genera ningún tipo de proceso (Lavoe, 2009).

2.3.2.3.3 Ventajas

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados (Lavoe, 2009).

2.4 Definición de sistemas de administración de base de datos

2.4.1 SQLite

SQLite es una biblioteca escrita en lenguaje C que implementa un sistema de gestión de bases de datos transaccionales SQL auto-contenido, sin servidor y sin configuración. El código de SQLite es de dominio público y libre para cualquier uso, ya sea comercial o privado. Actualmente es utilizado en gran cantidad de aplicaciones incluyendo algunas desarrolladas como proyectos de alto nivel, de manera que es la elección perfecta para crear bases de datos en sistemas operativos para móviles como Android o iOS (Marroquín, 2013).

2.4.1.1 Características de SQLite

- SQLite es un sistema completo de bases de datos que soporta múltiples tablas, índices, triggers y vistas. No necesita un proceso separado funcionando como servidor ya que lee y escribe directamente sobre archivos que se encuentran en el disco duro. El formato de la base de datos es multiplataforma e indistintamente se puede utilizar el mismo archivo en sistemas de 32 y 64 bits.
- La base de datos se almacena en un único fichero a diferencia de otros DBMS que hacen uso de varios archivos. SQLite emplea registros de tamaño variable de forma tal que se utiliza el espacio en disco que es realmente necesario en cada momento.

- El código fuente está pensado para que sea entendido y accesible por programadores promedio. Todas las funciones y estructuras están bien documentadas.
- Existe un programa independiente de nombre SQLite que puede ser utilizado para consultar y gestionar los ficheros de base de datos SQLite. También sirve como ejemplo para la escritura de aplicaciones utilizando la biblioteca SQLite (Marroquín, 2013).

2.4.1.1.1 Tipos de datos

SQLite usa un sistema de tipos inusual. En lugar de asignar un tipo a una columna como en la mayor parte de los sistemas de bases de datos SQL, los tipos se asignan a los valores individuales.

Cada uno de los datos almacenados en una base de datos SQLite implementa alguno de los siguientes tipos:

- NULL, un valor nulo.
- INTEGER, un entero con signo que se almacena en 1, 2, 3, 4, 5, 6 o 8 bytes de acuerdo a la magnitud del valor.
- REAL, un número de coma flotante (real), almacenado en 8 bytes.
- TEXT, una cadena de texto almacenada con las codificaciones UTF-8, UTF-16-BE o UTF-16-LE.
- BLOB, datos en formato binario, se almacenan exactamente como se introdujeron (Marroquín, 2013).

2.4.1.1.2 Ventajas de SQLite

- **Tamaño:** SQLite tiene una pequeña memoria y una única biblioteca es necesaria para acceder a bases de datos, lo que lo hace ideal para aplicaciones de bases de datos incorporadas.
- **Rendimiento de base de datos:** SQLite realiza operaciones de manera eficiente y es tan rápido como MySQL y PostgreSQL.
- **Portabilidad:** Se ejecuta en muchas plataformas y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración.

- **Estabilidad:** SQLite es compatible con un conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción (ACID), reunión de los cuatro criterios de atomicidad, consistencia, aislamiento y durabilidad.
- **SQL:** Implementa un gran subconjunto de la ANSI – 92 SQL estándar, incluyendo sub-consultas, generación de usuarios, vistas y triggers.
- **Interfaces:** Cuenta con diferentes interfaces del API, las cuales permiten trabajar con C++, PHP, Perl, Python, Ruby, un lenguaje de script que ha sido concebido con una sintaxis sencilla para facilitarse su aprendizaje (Tcl), Groovy, una biblioteca multiplataforma ampliamente usada para desarrollar aplicaciones con interfaz gráfica de usuario (Qt) ofrece el plugin sqlite, etc.
- **Costo:** SQLite es de dominio público, y por tanto, es libre de utilizar para cualquier propósito sin costo y se puede redistribuir libremente (Marroquín, 2013).

2.5 Publicación de contenidos

2.5.1 Fuente de noticias sindicalizadas (RSS)

Un RSS es un archivo generado por algunos sitios web que contiene una versión específica de la información publicada en esa web. Cada elemento de información contenido dentro de un archivo RSS se llama ítem (Daybis, 2014). Cada ítem consta normalmente de un título, un resumen y un enlace o URL a la página web de origen o que contiene el texto completo. Además puede contener información adicional como la fecha de publicación o el nombre del autor del texto.

El archivo RSS se reescribe automáticamente cuando se produce alguna actualización en los contenidos del sitio web. Accediendo al archivo RSS es posible saber si se han actualizado los contenidos y con qué noticias o textos, pero sin necesidad de acceder al sitio web salvo para leer la versión extendida (Daybis, 2014).

2.5.2 Canal de noticias electrónico

Un canal o recopilador de contenidos (también conocido como agregador de noticias o lector de RSS) es un tipo de software para suscribirse a fuentes de noticias en formatos RSS, Atom y otros derivados de XML como RDF/XML. El agregador reúne las noticias o historias publicadas en los sitios con redifusión web elegidos, y

muestra las novedades o modificaciones que se han producido en esas fuentes web; es decir, avisa de qué webs han incorporado contenido nuevo desde nuestra última lectura y cuál es ese contenido. Esta información es la que se conoce como fuente web. Un agregador es muy similar en sus presentaciones a los anteriores lectores de noticias (client newsreaders/NNTP), pero la tecnología XML y el web semántico los ha hecho más populares. Hoy en día, una enorme cantidad de blogs y sitios web ofrecen sus actualizaciones, que pueden ser fácilmente reunidas y administradas en un solo punto. Uno de los problemas con los agregadores de noticias es que el volumen de los artículos a veces puede ser abrumador, sobre todo cuando el usuario tiene muchas suscripciones web. Como solución, muchos de los lectores de feeds permiten a los usuarios etiquetar cada fuente con una o más palabras clave que pueden utilizarse para ordenar y filtrar los artículos disponibles en categorías fácilmente navegables. Otra opción es la importación del perfil de atención del usuario (en lenguaje APML, formato basado en XML para la captura de lo que le gusta (o los intereses) y lo que no a una persona) para filtrar los elementos según su relevancia para los intereses del usuario. Los lectores de feeds pueden ser aplicaciones web o aplicaciones locales que se instalan en el ordenador o dispositivo móvil del usuario (Daybis, 2014).

2.5.2.1 Políticas de publicación

La administración del aplicativo web está a cargo del webmaster del portal del Grupo Salesiano de Comunicación, las fuentes de noticias correspondientes a las diferentes unidades enlazadas a éste son responsabilidad de las mismas. El director de cada una de las unidades administrativas del Grupo Salesiano de Comunicación con presencia en el aplicativo móvil es responsable de la información que publica, y designará a una persona como responsable operativo y contacto con el departamento de sistemas. El Grupo Salesiano de Comunicación no es responsable de la información contenida en las fuentes externas enlazadas y tampoco respalda las opiniones que expresan (EdiBosco, 2013).

En base a las políticas de publicación y de contenido de noticias que maneja la Inspectoría Salesiana, se ha considerado la siguiente norma.

- La información será de utilidad para la comunidad en general y no podrá estar en desacuerdo con los principios de la empresa, que están consignados en los estatutos y reglamentos (EdiBosco, 2013).

En las noticias de primer plano podrán publicarse:

1. Noticias de carácter institucional.
2. Noticias de las dependencias que se generan por primera vez.
3. Noticias de actividades: culturales y religiosas.
4. Todos los días se publicarán entre una y cinco noticias principales que se consideren relevantes para el público. Dichas noticias deberán llevar una ampliación.
5. Si una noticia es publicada en el primer plano puede volverse a publicar allí siempre y cuando sea de gran relevancia y tenga nuevo contenido noticioso (EdiBosco, 2013).

2.5.2.2 Políticas de contenido

Para ser publicadas, todas las noticias deberán contener obligatoriamente:

- 1. Título:** Debe estar en formato negrita y contener de 5 a 15 palabras.
- 2. Descripción:** Debe contener máximo 1800 caracteres en función y dependencia de las imágenes; tanto el título como la descripción deben preservar la buena redacción y ortografía, siguiendo las reglas sintácticas, gramáticas y ortográficas del idioma español utilizadas en Ecuador.
- 3. Imagen:** Las fotografías adicionales que apoya la información no son de carácter obligatorio. Las imágenes a publicar deben editarse con tamaños apropiados que no alteren el diseño de la aplicación móvil, se deben usar imágenes en formato JPEG, PNG o GIF con tamaños no mayores a 400×400 píxeles. Los elementos gráficos de animación que parpadeen, se desplacen o se actualicen automáticamente deben tener la posibilidad de ser detenidos temporal y totalmente. En el caso de noticias con pocos caracteres se compensará con las imágenes existentes, solo se usará una

imagen pero hay un máximo permitido de dos imágenes para no saturar la visualización de la noticia.

4. Estilo: Las noticias deberán estar redactadas bajo los parámetros de un medio digital, de manera que permitan una navegación rápida y ágil. Además todas las noticias serán sometidas a revisión de un corrector de estilo antes de su edición (EdiBosco, 2013).

2.5.2.2.1 Información

1. Los textos institucionales, de cualquier índole y objetivo deben conservar parámetros de calidad en la medida en que han de ser fidedignos, confiables, inteligibles, vigentes, puntuales, indubitables, accesibles, verificables y válidos.
2. La información publicada deberá ser de interés para el público que accede al medio (EdiBosco, 2013).

2.5.2.2.2 Extensión

1. Todas las noticias publicadas deberán tener un autor con el nombre de un contacto y con la fecha de actualización.
2. En los casos que se considere pertinente se hará uso de hipervínculos para evitar exceso de texto.
3. Cada noticia podrá permanecer publicada máximo 3 meses, una vez cumplido este plazo será retirada del medio por el webmaster.
4. Será deber de quien envíe una noticia, informar en caso de que el evento o actividad haya sido cancelado o aplazado (EdiBosco, 2013).

CAPÍTULO 3

ANÁLISIS Y DISEÑO

En este capítulo se indica los requerimientos no funcionales y funcionales del aplicativo móvil, además se explica de forma detallada cada uno de los diagramas utilizados como parte del diseño de la solución.

3.1 Obtención de requerimientos

El proceso para la recolección de requerimientos es el más importante en lo que se refiere al desarrollo de software, también llamado ingeniería de requerimientos (Álvarez, 2009), porque permite gestionar las necesidades del proyecto en forma estructurada y mejora la calidad del software, este proceso comprende las actividades de obtención, análisis, especificación y validación de requerimientos. Existen varias técnicas que son utilizadas al momento de realizar los pasos indicados, muchas de ellas son utilizadas también en otras áreas (Yorio, 2011). A continuación se muestran las técnicas y el modelamiento que se tomaron en cuenta para este proyecto.

- **Encuesta:** Sirve para obtener información, hecha de manera escrita en papel u otro medio a un grupo de personas (Álvarez, 2009). Se realizó una encuesta a 20 personas, entre empleados y clientes de la editorial, para conocer su criterio acerca del detalle de los servicios que ofrece el Grupo Salesiano de Comunicación y para identificar los inconvenientes al momento de consultar las noticias (Ver Anexo 1).

- **Observación:** Mediante esta actividad se conoce los procesos relacionados con el problema sin obstruir a los usuarios en sus tareas cotidianas (Álvarez, 2009). Durante el proceso de recolección de requerimientos se identificó los procedimientos y características de los usuarios permitiendo conocer los flujos de información.

- **Diagramas:** Es una representación gráfica del flujo de datos entre las tareas que se realizan en el sistema (Álvarez, 2009). Según la metodología Mobile-D se tomaron en cuenta los diagramas de casos de uso, de bloques, de clases y de navegabilidad; con estos diagramas se identificó tanto tareas como actores externos.

Según lo mencionado anteriormente, primero se realizaron actividades de obtención de requerimientos a través de una encuesta, luego se realizó el análisis de los requerimientos funcionales y no funcionales basándose en las etapas de la

metodología Mobile-D planteada para este proyecto, después se realizó la especificación y validación de requerimientos utilizando diagramas.

- Modelo de requisitos

El modelo de requisitos estructura los requerimientos de usuario final, los cuales se dividen en requisitos no funcionales y funcionales. Tanto los primeros como los segundos definen la expectativa del usuario frente al sistema. Lo que se pretende con este modelo de requisitos es que el aplicativo móvil funcione correctamente y muestre noticias actualizadas y en tiempo real a los usuarios, permitiéndoles estar informados de las novedades que se susciten dentro del Grupo Salesiano de Comunicación.

3.1.1 Requerimientos no funcionales

Estos requerimientos describen cualidades del sistema que no son esencialmente aspectos funcionales, aunque son importantes en el contexto de uso de la aplicación, los cuales fueron definidos conjuntamente con el cliente. Para este proyecto se han tomado en consideración los siguientes requerimientos no funcionales.

Rendimiento: Define el tiempo de respuesta en la ejecución del aplicativo móvil y se ha tomado en cuenta las siguientes métricas.

Métricas y/o medios de verificación

- El uso de un protocolo de comunicación rápido y eficiente es importante para que la respuesta del sistema sea rápida.
- El tiempo de respuesta promedio al momento de ejecutar la aplicación móvil no debe superar los 2 segundos.

Seguridad: Define la integridad y la privacidad de la información almacenada tanto en el aplicativo móvil como en la base de datos y se ha tomado en consideración la siguiente métrica.

Métrica y/o medio de verificación

- Se debe usar un servicio web ya que utiliza un fuerte sistema de seguridad en los firewalls porque se encuentra apoyado en la tecnología HyperText Transfer Protocol (HTTP).

Cabe mencionar que en la aplicación móvil no habrá autenticación de usuario, lo que significa que no se requerirán credenciales de usuario, debido a que ésta no almacenará información que pueda ser susceptible a alteraciones, es decir, las noticias generadas dentro del Grupo Salesiano de Comunicación serán de dominio público y no habrá restricción en su contenido por lo que cualquier persona tendrá acceso a ellas.

Fiabilidad: Define la probabilidad de que el aplicativo móvil funcione adecuadamente durante un período determinado bajo condiciones operativas específicas y se ha tomado en cuenta la siguiente métrica.

Métrica y/o medio de verificación

- Que el sistema sea 90% fiable, es decir, que la aplicación de cada 10 ejecuciones las 9 no tengan ningún error, que la información sea fidedigna y que no exista ningún error de sintaxis (muestra caracteres especiales).

Disponibilidad: Define el porcentaje de tiempo que el aplicativo móvil puede ser usado en trabajo productivo y se ha tomado en consideración la siguiente métrica.

Métrica y/o medio de verificación

- En funcionamiento normal, el sistema estará disponible el 100% del tiempo; esto se obtiene instalando el aplicativo dentro del dispositivo móvil como una aplicación nativa y estableciendo una conexión a Internet para cargar información actualizada y en tiempo real.

Mantenibilidad: Define la cantidad de esfuerzo requerido para conservar el funcionamiento normal del aplicativo móvil a los cambios presentados durante las versiones o para restituirlo una vez que se ha presentado un evento de falla y se ha tomado en cuenta la siguiente métrica.

Métrica y/o medio de verificación

- Cada uno de los componentes de software que forman parte de la solución propuesta deberán estar debidamente documentados como respaldos para dar mantenimiento al software.

Portabilidad: Define la característica que posee el aplicativo móvil para ejecutarse en diferentes plataformas, el código fuente del software es capaz de reutilizarse en

vez de crearse un nuevo código cuando el software pasa de una plataforma a otra y se han tomado en consideración las siguientes métricas.

Métricas y/o medios de verificación

- Compatibilidad con plataformas: El sistema desarrollado ofrecerá compatibilidad con otras plataformas Android desde la versión 2.2.
- La aplicación móvil podrá descargarse desde cualquier parte del mundo gratuitamente por medio de la tienda Google Play.

Usabilidad: Define la facilidad que los usuarios tienen para interactuar con el aplicativo móvil y se han tomado en cuenta las siguientes métricas.

Métricas y/o medios de verificación

- La aplicación móvil debe seguir la figura corporativa de Editorial Don Bosco.
- Es conveniente que los elementos pulsables de la interfaz sean lo suficientemente grandes para que sea cómodo tocarlos.
- Es adecuado emplear un tamaño de fuente grande para garantizar la legibilidad del texto.
- Es recomendable que las búsquedas alfabéticas de una palabra clave ignoren la diferencia entre mayúsculas y minúsculas.

3.1.1.1 Interfaces de usuario

Según el análisis realizado, el aplicativo móvil tendrá las siguientes interfaces de usuario (secciones):

1. Sección Índice: Está conformada por un panel desplegable por medio de la función touch (acción sensible al toque de un lápiz especial o a los dedos) que permite elegir dos grupos de selección de contenidos, los cuales son:

- **Funciones de filtrado y acceso rápido:** Permite desplegar noticias basadas en búsquedas o grupos de interés y tiene las siguientes opciones:
 - **Entradas nuevas:** Es un acceso directo al listado de noticias nuevas, además cuenta con un contador de las mismas fácil de identificar por el usuario.

- **Favoritos:** Es un acceso directo al listado de las noticias elegidas por el usuario como favoritas por medio del ícono en forma de estrella presente en cada una de las noticias presentadas por la aplicación, además cuenta con un contador de las mismas fácil de identificar por el usuario.
- **Búsqueda mediante un texto:** Permite realizar búsquedas mediante texto en todos los grupos de noticias mostrando los resultados a manera de lista e indicando el grupo al que pertenece cada una de las noticias filtradas.
- **Obras salesianas:** Permite visualizar un listado ordenado de todas las obras salesianas contempladas en este proyecto con sus respectivos subgrupos, éste permite dirigirse a los canales ordenados de las obras respectivas, cada subgrupo cuenta con un contador de noticias que no han sido abiertas por el usuario, además de indicar el tiempo de la última vez que fue actualizada cada uno de los grupos de noticias.

Cabe mencionar que se pueden agregar nuevos tópicos de noticias afines al Grupo Salesiano de Comunicación, si éste lo considera necesario, en la sección Índice.

2. Sección Ubicación: Permite identificar rápidamente en que área de contenido de la aplicación se encuentra el usuario.

3. Sección Funciones: Muestra varias funciones que pueden ser aplicadas a la información del contenido actual ya sea en las listas o en el panel de lectura de una noticia. Estas opciones son:

- Funciones en listas de noticias.
 - Ocultar lectura de entradas: Oculta las noticias leídas.
 - Refrescar contenido: Carga nuevas noticias.
 - Mostrar todos como leídos: Marca todas las noticias como leídas.
- Funciones en panel de lectura de noticias
 - Marcar como favorito: Marca una noticia como favorita (elegida por el usuario).
 - Compartir: Comparte una noticia en diferentes redes sociales instaladas en el dispositivo móvil (Facebook, Twitter, etc.).

- Pantalla completa: Amplía a pantalla completa la visualización de una noticia.
- Mostrar texto completo: Muestra todo el detalle de una noticia.
- Mostrar en el navegador: Muestra una noticia en el navegador instalado en el dispositivo móvil (Mozilla Firefox, Google Chrome, etc.).
- Traducir en la Web: Traduce una noticia al idioma inglés.

Cabe mencionar que al elegir una noticia, el usuario podrá administrar la misma, según sus necesidades.

4. Sección Contenido: En esta área se encuentra la información noticiosa como tal en distintos formatos, estos formatos son:

- **Lista:** Listado donde indica todas las noticias existentes de una obra o filtro seleccionado. Dichas listas permitirán realizar funciones rápidas en cada uno de los ítems, estas funciones son, marcar noticias como leídas o no y también seleccionar noticias como favoritas o no. Adicional a esto se puede visualizar a que grupo pertenecen cada una de las noticias, además de la hora y fecha de la publicación de las mismas.
- **Individual:** Además de mostrar información noticiosa, el área de contenidos permite desplegar ya sean listas de configuración o información predeterminada de la aplicación.

5. Sección Configuración: Muestra un listado de opciones administrativas que tendrán efecto global sobre la información desplegada en la sección Contenido. Estas opciones son las siguientes:

- Refresco automático
 - Activar refresco automático: Activa el refresco automático de noticias.
 - Intervalo de refresco: Muestra el intervalo entre actualizaciones automáticas.
 - Refrescar noticias al abrir: Refresca todas las noticias al abrir la aplicación.

- Refrescar solo con red WiFi: Refresca automáticamente cuando la red WiFi está activa.
- Presentación del contenido
 - Aspecto: Cambia el color del tema de la interfaz de la aplicación móvil.
 - Tiempo que se mantendrán las noticias en la lista: Muestra la fecha que se mantendrán las noticias antes de ser borradas.
 - Tamaño de texto: Cambia el tamaño del texto del contenido.
 - Desactivar imágenes: No muestra ni carga ninguna imagen.
 - Descargar imágenes: Descarga todas las imágenes en el almacenamiento local.
 - Orden de las noticias: Muestra noticias antiguas primero.
- Notificaciones
 - Activar notificaciones: Activa notificaciones en caso de nuevas noticias.
 - Alerta de audio: Selecciona la alerta de audio ante nuevas noticias.
 - Vibrar: Activa la vibración del dispositivo móvil ante nuevas noticias.

6. Sección Accesos Directos: Esta sección se encuentra conformada por widgets, nativos de la plataforma Android (detallado en el punto 2.1.2), que permiten colocar mini aplicaciones en las pantallas o escritorios principales de la plataforma. En este caso se tiene la posibilidad de agregar dos tipos de widgets los cuales son:

- **Contador de nuevas noticias:** Muestra un número donde se indica la cantidad de nuevas noticias sin leer, y permite abrir la aplicación al hacer un touch sobre la misma.
- **Listado de noticias:** Permite mostrar un listado sencillo de todas las noticias o de las que el usuario decida que son de su preferencia. Y permite abrir la aplicación al hacer un touch sobre la misma. Además muestra funciones como el color de fondo de widget y el tamaño del texto a mostrar.

7. Sección Notificaciones: Esta sección se encuentra presente en el panel nativo de notificaciones de la plataforma, y permite informar al usuario si existen nuevas entradas noticiosas.

A partir de este análisis se pueden plantear los controles a utilizar en la pantalla principal, cuyo objetivo es mostrar: Un panel desplegable con las obras salesianas (Grupo Salesiano de Comunicación, Editorial Don Bosco, Librerías LNS, Audiovisuales Don Bosco, Centro Salesiano de Publicaciones Pastorales José Ruaro y Centro Gráfico Salesiano). A continuación la lista de controles gráficos:

- La pantalla principal, de color negro con letras blancas tanto en la parte superior como en el panel desplegable, tiene un solo elemento raíz que es la sección Índice; la lista de noticias es una interfaz, la cual es un componente tipo contenedor que muestra una serie de ítems de color blanco en su entorno y letras negras. Este elemento ocupa toda la pantalla; el contenido de cada uno de estos ítems presenta un título, una imagen y un detalle.
- El ítem es un compuesto de varios controles; este ítem en este caso es la noticia; a continuación la composición de esta opción:
 - **Título de la noticia:** Se muestra una etiqueta grande con letras de color negro y entorno de color blanco que contiene como texto el título de la noticia.
 - **Imagen:** Se muestra una ilustración debajo del título de la noticia.
 - **Descripción de la noticia:** Se muestra una etiqueta debajo de la imagen, con letras de color negro y con un texto más opaco y pequeño que el título.
 - **Botones de función externa:** Se muestra dos opciones con letras de color negro y entorno de color celeste claro donde el usuario selecciona lo siguiente:
 - Mostrar texto completo.
 - Mostrar en el navegador.
 - Traducir en la Web.

3.1.1.2 Interfaces de software

Se implementará sobre la plataforma Android, para lo cual se deberá descargar el aplicativo desde la tienda Google Play a través del servicio de Internet como medio de acceso, por lo que es necesario contratar un paquete de datos de una operadora de telefonía móvil o conectarse a una red WiFi.

El aplicativo móvil consumirá datos del sitio web informativo de la editorial porque éste refleja la información que existe en la página web y así se obtienen datos actualizados y en tiempo real, los cuales son mostrados al usuario a través del aplicativo móvil, es por eso que se necesita de un servicio web para poder obtener la información que se encuentra en la base de datos que utiliza el sitio web, ya que, permite a la empresa intercambiar datos sin necesidad de conocer los detalles de sus respectivos sistemas de información y no proveen al usuario una interfaz gráfica (GUI). En vez de ello, comparten la lógica del negocio, los datos y los procesos a través de la red.

3.1.1.3 Interfaces de comunicaciones

El sistema será accedido a través de dispositivos móviles conectados al servicio de Internet sobre el protocolo de comunicaciones HyperText Transfer Protocol (HTTP) de forma que permite a todos los usuarios realizar transacciones mediante el acceso al aplicativo móvil (Gutierrez, 2010). El protocolo de comunicaciones antes mencionado, es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente en este caso el usuario del aplicativo móvil y un servidor es decir el lugar donde residirá o estará la aplicación móvil.

3.1.2 Requerimientos funcionales

La formulación de los requerimientos funcionales constituye el primer paso en el diseño de un sistema; es decir se manifiesta las características esenciales del proyecto Análisis, Diseño, Construcción e Implementación de un aplicativo móvil para Administración y Distribución de noticias, de manera que sirva de guía para determinar los casos de uso, los mismos que especifican el comportamiento del sistema a desarrollar. Los requisitos funcionales han sido registrados en tablas, cuyos campos se detallan a continuación:

- **Código:** Identificador del requisito funcional. El formato definido es: ERF-001 (número secuencial).

- **Nombre del requerimiento:** Funcionalidad del aplicativo.
- **Actor:** Usuario o proceso del sistema.
- **Tipo de requerimiento:** Clasificación.
- **Entrada:** Datos, información o salida de otros procesos, los cuales producirán la funcionalidad requerida.
- **Salida:** Resultado o garantía de éxito en el cumplimiento del requerimiento funcional.
- **Detalle:** Texto que describe la funcionalidad requerida en el software y contiene lo siguiente:
 - Objetivo
 - Descripción
 - Escenario
- **Observaciones:** Información adicional complementaria.

Tabla 1

Mostrar al usuario los grupos de noticias de las obras salesianas de forma ordenada.

Código: ERF-001	Requerimiento: Mostrar al usuario los grupos de noticias de las obras salesianas de forma ordenada.	
Actores: Usuario, aplicativo móvil.		Tipo de requerimiento: Funcional
Entrada: Abrir aplicación móvil a través de la función touch.		Salida: Mostrar un listado ordenado y agrupado de las noticias del Grupo Salesiano de Comunicación.
Detalle		
Objetivo Permitir al usuario visualizar los grupos de noticias de las obras salesianas de forma ordenada.		
Descripción: Seleccionar un grupo de noticias.		
Escenario: El usuario debe abrir la aplicación móvil, ubicarse en el menú principal (sección Índice), luego buscar una obra salesiana y seleccionar la misma para visualizar el resultado de la selección.		
Observaciones: El usuario debe desplazarse en la sección Índice con la función touch para visualizar todo el contenido.		

Nota: Requerimiento funcional

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 2

Permitir al usuario conocer en que sección del aplicativo móvil se encuentra.

Código: ERF-002	Requerimiento: Permitir al usuario conocer en que sección del aplicativo móvil se encuentra.	
Actores: Usuario, aplicativo móvil.		Tipo de requerimiento: Funcional
Entrada: Abrir aplicación móvil a través de la función touch y seleccionar cualquier sección.		Salida: Visualizar una barra con información de la ubicación dentro de la aplicación móvil.
Detalle		
Objetivo: Permitir al usuario conocer en que sección se encuentra dentro de la aplicación móvil.		
Descripción: Mostrar la ubicación al usuario.		
Escenario: El usuario debe abrir la aplicación móvil, realizar una selección de cualquier sección y visualizar en que sección se encuentra dentro del aplicativo móvil.		
Observaciones: La barra de ubicación está siempre visible en todas las secciones para ser visualizada por el usuario.		

Nota: Requerimiento funcional

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 3

Brindar al usuario un conjunto de funciones que permitan visualizar, compartir, ordenar y refrescar las noticias.

Código: ERF-003	Requerimiento: Brindar al usuario un conjunto de funciones que permitan visualizar, compartir, ordenar y refrescar las noticias.	
Actores: Usuario, aplicativo móvil.		Tipo de requerimiento: Funcional
Entrada: Seleccionar un grupo de noticias del Grupo Salesiano de Comunicación a través de la función touch.		Salida: Visualizar una barra de íconos con funciones que se aplican a noticias individuales o a un grupo de noticias.
Detalle		
Objetivo: Brindar al usuario varias funciones que se aplican tanto a noticias individuales como a un grupo de noticias.		
Descripción: Visualizar, compartir, ordenar y refrescar la información de las noticias.		
Escenario: El usuario debe seleccionar un grupo de noticias del Grupo Salesiano de Comunicación, elegir una noticia y seleccionar una función sobre el contenido de la misma.		
Observaciones: Las funciones solo se aplicarán si el usuario las elige, caso contrario solo se muestra la información noticiosa. Además las funciones quedan grabadas en el aplicativo móvil para el próximo ingreso.		

Nota: Requerimiento funcional

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 4

Mostrar al usuario los contenidos de las noticias de manera ordenada.

Código: ERF-004	Requerimiento: Mostrar al usuario los contenidos de las noticias de manera ordenada.	
Actores: Usuario, aplicativo móvil.		Tipo de requerimiento: Funcional
Entrada: Abrir aplicación móvil a través de la función touch y seleccionar cualquier sección.		Salida: Visualizar la información noticiosa en la sección Contenido.
Detalle		
Objetivo: Permitir que el usuario lea el contenido de las noticias de manera ordenada.		
Descripción: Visualizar el contenido de las noticias.		
Escenario: El usuario debe abrir la aplicación móvil, realizar una selección de cualquier sección y visualizar el contenido de las noticias.		
Observaciones: El usuario debe desplazarse en la sección Índice con la función touch para seleccionar una obra salesiana y visualizar la información noticiosa en la sección Contenido.		

Nota: Requerimiento funcional

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 5

Brindar al usuario un listado de configuraciones aplicables a la información noticiosa.

Código: ERF-005	Requerimiento: Brindar al usuario un listado de configuraciones aplicables a la información noticiosa.	
Actores: Usuario, aplicativo móvil.		Tipo de requerimiento: Funcional
Entrada: Abrir aplicación móvil a través de la función touch.		Salida: Visualizar cambios realizados al activar o desactivar las configuraciones.
Detalle		
Objetivo: Permitir que el usuario seleccione varias opciones del listado de configuraciones.		
Descripción: Activar o desactivar configuraciones.		
Escenario: El usuario debe abrir la aplicación móvil, elegir la sección Configuración, activar o desactivar las configuraciones disponibles y visualizar los cambios realizados.		
Observaciones: El usuario debe seleccionar la sección Configuración con la función touch para activar o desactivar las configuraciones disponibles, éstas solo se aplicarán si el usuario las elige.		

Nota: Requerimiento funcional

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 6

Brindar al usuario accesos directos para visualización e ingreso de nuevas noticias.

Código: ERF-006	Requerimiento: Brindar al usuario accesos directos para visualización e ingreso de nuevas noticias.	
Actores: Usuario, dispositivo móvil, aplicativo móvil.		Tipo de requerimiento: Funcional
Entrada: Agregar el widget de la aplicación móvil al escritorio de Android.		Salida: Elegir widget de contador o de listado de noticias.
Detalle		
Objetivo: Permitir que el usuario acceda de manera directa a la aplicación móvil a través de accesos directos.		
Descripción: Acceder al aplicativo móvil de manera directa y visualizar las nuevas noticias existentes.		
Escenario: El usuario debe elegir la opción de agregar widget al escritorio de Android, seleccionar el widget del aplicativo móvil y elegir el widget ya sea de contador o listado de noticias.		
Observaciones: El usuario tiene la potestad de seleccionar el widget que le permita acceder de una manera más rápida a la aplicación móvil.		

Nota: Requerimiento funcional

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 7

Brindar al usuario alertas o notificaciones de nuevas noticias.

Código: ERF-007	Requerimiento: Brindar al usuario alertas o notificaciones de nuevas noticias.	
Actores: Usuario, dispositivo móvil, aplicativo móvil, servicio web.		Tipo de requerimiento: Funcional
Entrada: Configurar el tiempo de sincronización de las noticias a través de la función touch.		Salida: Visualizar las notificaciones.
Detalle		
Objetivo: Permitir que el usuario configure el tiempo de sincronización de las noticias.		
Descripción: Configurar el tiempo de sincronización de la información noticiosa.		
Escenario: El usuario debe configurar el tiempo de sincronización de las noticias, visualizar la notificación y seleccionar la misma.		
Observaciones: Las alertas aparecerán en el panel de notificaciones del dispositivo móvil con el número de nuevas noticias existentes.		

Nota: Requerimiento funcional

Elaborado por: Edwin Ceballos / Francisco Miño

3.1.2.1 Restricciones

Las características mínimas para el buen funcionamiento del aplicativo móvil son:

- Android 2.2. (Froyo).
- Memoria RAM 512 MB.
- Conexión a Internet 2G³ o WiFi.

Las características óptimas para el buen funcionamiento de la aplicación son las siguientes:

- Android 4.1.x (Kit Kat) o superior.
- 1 GB de RAM.
- 8 GB de memoria interna.
- Conexión a Internet 3G, HSPA+, Lte⁴ o WiFi.

3.2 Análisis

3.2.1 Propósito

Con el análisis se pretende definir correctamente los requerimientos del sistema, es decir comprender con exactitud el problema al que se le va a proporcionar una solución automatizada, es por eso que se necesita utilizar el modelo de calidad ISO 9126, éste define las características de calidad a usar en la evaluación de este producto informático de tipo software y se basará en el estándar IEEE Recommended Practices for Requirements Specification ANSI/IEEE st. 830, 1998; el cual sirve para abstraer principalmente los conceptos funcionales del producto que se espera realizar, ayuda a validar y completar la lista de requerimientos de software, ayuda a identificar los tipos de pruebas a realizar y a determinar criterios de aceptación del software; este modelo de calidad y este estándar dan las pautas necesarias para utilizar la especificación de requisitos de software (ERS) para el desarrollo del aplicativo móvil.

³ Notifica que la conexión es lenta. Puede influir en la carga de noticias pero no significa que la aplicación móvil no funcione.

⁴ Notifica que no existe ninguna restricción porque son conexiones de alta velocidad y el aplicativo no se ve afectado por las mismas.

3.2.2 Alcance

Con el análisis se procura llegar a desarrollar un aplicativo móvil para la plataforma Android que recepte información de la aplicación web, existente en la editorial y que provea un buen tiempo de respuesta, es decir, se determina tanto la rapidez al momento de ejecutar la aplicación móvil como la realización de una tarea en condiciones particulares de trabajo, para lo cual se definirán procesos y entes que intervendrán en esta actividad, esperando mantener en todo momento un control eficaz de los recursos, logrando así presentar de una manera rápida las noticias generadas dentro del Grupo Salesiano de Comunicación; y se ha determinado lo siguiente:

- El sistema está destinado a mostrar noticias únicamente en dispositivos móviles, no en computadoras personales.
- El sistema está destinado a tipos de dispositivos móviles que soporten imágenes y tengan acceso a Internet.

El sistema se identificará como EdiBosco Noticias, lo que equivale a Administración y Distribución de noticias para los usuarios de la Editorial Don Bosco.

3.2.3 Visión general del documento

Según la metodología Mobile-D, este documento sirve para tener una visión general del aplicativo móvil, definir las funciones principales, los factores que inciden en el sistema a nivel general y los requisitos que debe satisfacer la aplicación móvil, para luego evaluar el rendimiento del sistema a través de pruebas.

3.2.4 Descripción global

El aplicativo móvil EdiBosco Noticias mostrará al usuario noticias actualizadas y en tiempo real referentes a las actividades que se realicen dentro del Grupo Salesiano de Comunicación.

3.2.5 Perspectiva del producto

La aplicación móvil interactuará con la aplicación web por medio del servicio web, debe funcionar correctamente y brindar a los usuarios la posibilidad de ver noticias actuales en cualquier momento desde su dispositivo móvil, cabe mencionar que este aplicativo está dirigido a todas aquellas personas que posean un dispositivo móvil,

siempre y cuando éste cumpla con los requerimientos del sistema mencionados en el punto 3.1.2.1.

3.2.5.1 Ventajas

- Proveer acceso al usuario de información donde sea que se encuentre mientras disponga de conexión a Internet.
- El mercado de los servicios móviles se encuentra creciendo exponencialmente, y los que adapten la tecnología en sus fases iniciales serán los grandes ganadores.
- Una aplicación para dispositivo móvil puede utilizarse con fines de promoción del negocio, o ser vendida como producto de consumo.

3.2.5.2 Beneficios

- Extensión móvil de la empresa.
- Permite al usuario ponerse en contacto en cuestión de segundos.
- Provee un escaparate que puede ser visto por miles de personas al día.
- Es una herramienta de mercadotecnia de bajo costo.

3.3 Diagramación

Consiste en la representación gráfica de las tareas y de las relaciones entre dichas tareas a través de artefactos como los diagramas de casos de uso y de clases, basados en las fases que presenta la metodología utilizada en este proyecto, mencionada en el punto 2.2.1, la cual, según (Lasso, 2013) dice que son compatibles con el diseño de una aplicación móvil. Esencialmente se refiere a la organización de los contenidos del producto, al funcionamiento básico del mismo y la ubicación que tendrán estos contenidos en la interfaz (Olsen, 2007).

3.3.1 Diagramas de caso de uso

Los diagramas de casos de uso documentan el comportamiento del aplicativo móvil desde el punto de vista del usuario y se basan en los requerimientos funcionales antes mencionados, ya que estos manifiestan las características esenciales del aplicativo móvil EdiBosco Noticias. A continuación se muestra el siguiente diagrama de caso de uso:

Funcionamiento de aplicación móvil EdiBosco Noticias.

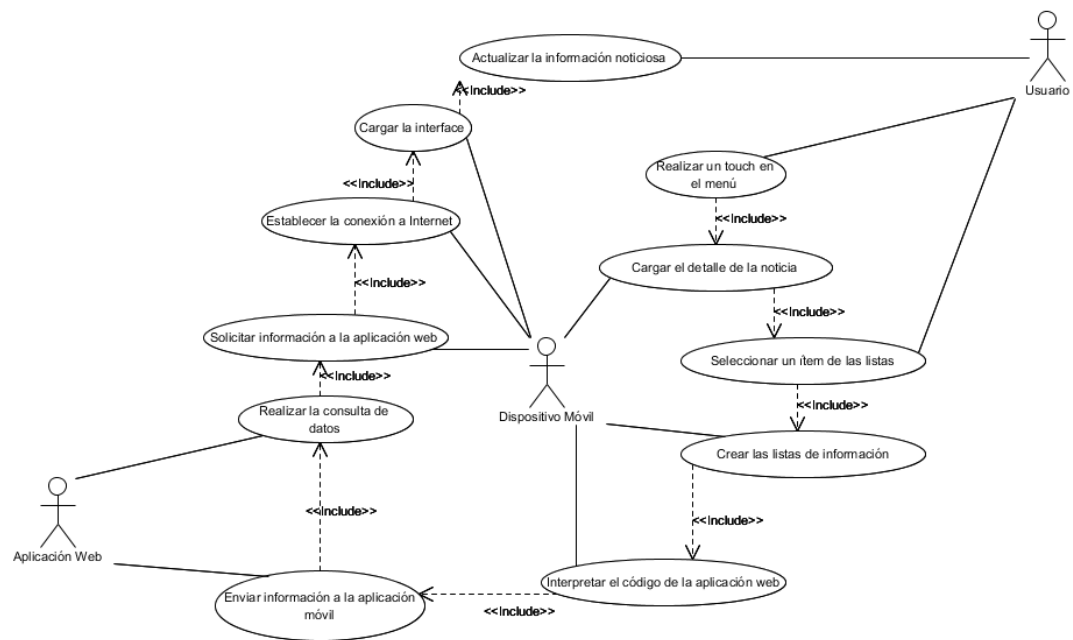


Figura 7

Elaborado por: Edwin Ceballos / Francisco Miño

Es necesario mencionar que en la relación `<<include>>` se refiere a que el primer caso de uso (base) incluye al segundo caso de uso. Es decir, el segundo es parte esencial del primero. Sin el segundo, el primero no podría funcionar bien; pues no podría cumplir su objetivo, en cambio en la relación `<<extends>>` se utiliza cuando un caso de uso base incorpora el comportamiento de otro caso de uso y extiende su funcionamiento.

Tabla 8

Actores principales que interactúan con el aplicativo móvil EdiBosco Noticias.

Actor	Usuario
Descripción	Usuario que utiliza la aplicación para dispositivo móvil con el fin de obtener información sobre las noticias que la Editorial Don Bosco genera, el cual realiza acciones sobre las mismas.
Tipo	Actor principal
Actor	Aplicación web
Descripción	Herramienta que el webmaster utiliza accediendo a un servicio web a través de Internet o de una intranet mediante un navegador.
Tipo	Actor principal
Actor	Dispositivo móvil
Descripción	Dispositivo construido sobre una plataforma informática móvil (en este caso Android), con una mayor capacidad de almacenar datos y realizar actividades semejantes a una minicomputadora.
Tipo	Actor principal
Actor	Webmaster
Descripción	Usuario que se encarga de administrar las noticias e interactúa con el sitio web informativo de la Editorial Don Bosco para realizar dicha gestión.
Tipo	Actor principal

Nota: Detalle general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 9

Tareas de caso de uso principal.

ACTOR	TAREA	DESCRIPCIÓN TAREA
Aplicación web	Realizar la consulta de datos	La aplicación web determina si la información es correcta.
	Enviar información a la aplicación	La aplicación web envía datos a la aplicación móvil para que el usuario visualice en forma de noticia.
Dispositivo móvil	Cargar la interface	Después que el usuario ingresa al aplicativo EdiBosco Noticias el dispositivo móvil carga la pantalla principal de la aplicación.
	Establecer conexión a Internet	El dispositivo móvil se conecta al Internet.
	Solicitar información a la aplicación web	El dispositivo móvil se sincroniza con la aplicación web para cargar datos.
	Interpretar el código de la aplicación web	La aplicación web envía información al aplicativo móvil y éste genera datos a ser visualizados por el usuario.
	Crear las listas de información	El dispositivo móvil carga la lista de obras salesianas y sus respectivas noticias.
	Cargar el detalle de la noticia	El dispositivo móvil muestra la noticia en la pantalla.
Usuario	Seleccionar ícono de la aplicación	El usuario ingresa al aplicativo móvil.
	Seleccionar un ítem de la lista	El usuario elige una obra salesiana y selecciona la noticia respectiva.
	Leer la noticia	El usuario observa la noticia.
Webmaster	Ingresar al sitio web informativo de la Editorial	El webmaster carga el sitio web informativo de la Editorial para administrar información.
	Administrar las noticias	El webmaster tiene privilegios para ingresar, editar, modificar y eliminar una noticia.

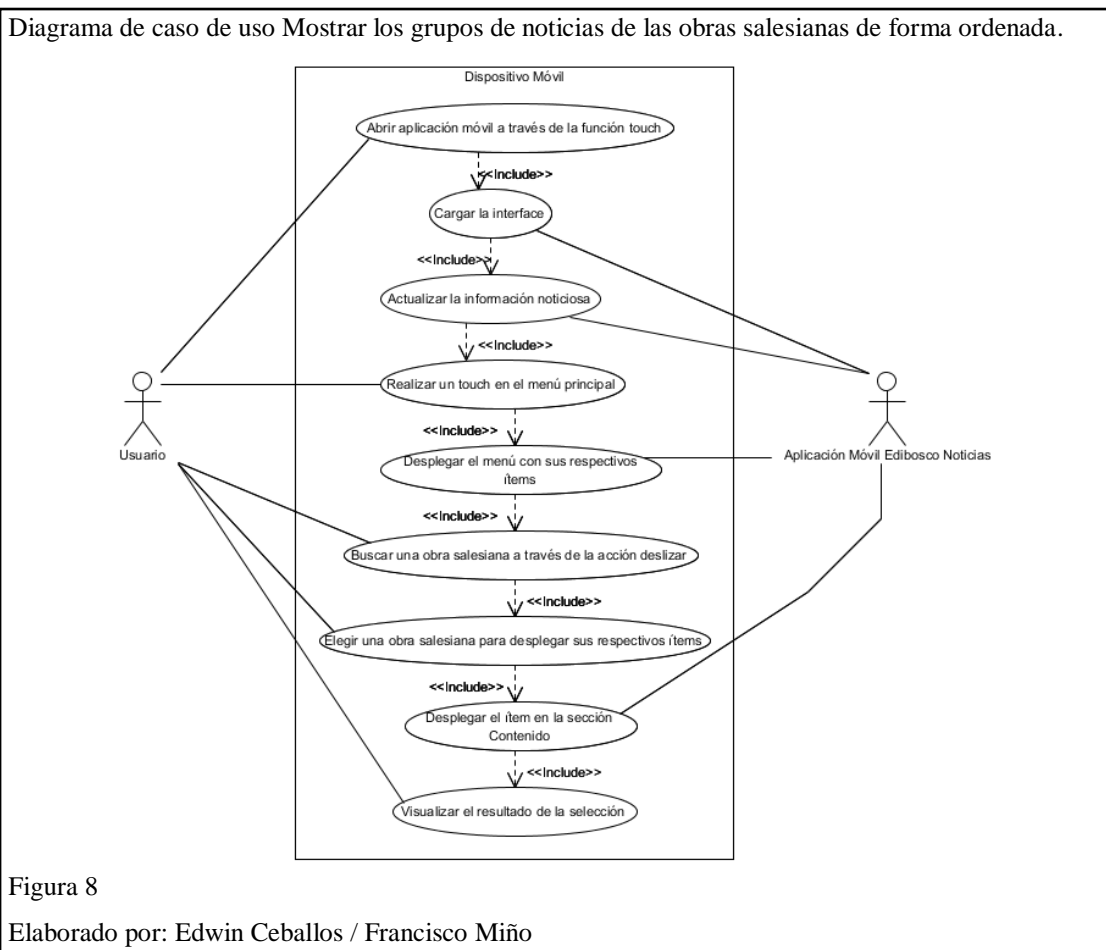
Nota: Detalle general

Elaborado por: Edwin Ceballos / Francisco Miño

Cada uno de los requerimientos funcionales del aplicativo móvil es representado con su propio diagrama de caso de uso, ya que su principal característica es describir los actores que intervienen en cada tarea y el detalle de las mismas; a continuación se muestran los siguientes diagramas de casos de uso:

- Tareas del usuario

Para el requerimiento funcional Mostrar los grupos de noticias de las obras salesianas de forma ordenada, se ha establecido lo siguiente:



Descripción de caso de uso: Mostrar los grupos de noticias de las obras salesianas de forma ordenada.

Actores que intervienen: Usuario, Aplicación móvil.

Una vez que el usuario contrate un paquete de datos de una operadora de telefonía móvil o se conecte a una red WiFi, éste podrá descargar a su dispositivo el aplicativo e interactuará (realizará un touch) con el objeto EdiBosco Noticias, permitiéndole ingresar al aplicativo móvil, luego seleccionará la sección Índice (menú principal) donde se encuentran estos ítems: Grupo Salesiano de Comunicación, Editorial Don Bosco, Librerías LNS, Audiovisuales Don Bosco, Centro Salesiano de Publicaciones Pastorales José Ruaro y Centro Gráfico Salesiano, elegirá el grupo de

noticias y a continuación se desplegará un listado de noticias donde seleccionará la noticia que desea visualizar.

Pre-Condición: Todo usuario que haga uso de esta aplicación, primero debe contratar un paquete de datos o conectarse a una red WiFi para descargar el aplicativo desde la tienda Google Play a su dispositivo móvil para luego abrir el mismo.

Acciones

- 1) El usuario abre el aplicativo EdiBosco Noticias a través de la función touch.
- 2) El usuario realiza un touch en el menú principal.
- 3) El usuario busca una obra salesiana a través de la acción deslizar.
- 4) El usuario elige una obra salesiana para desplegar los ítems correspondientes.
- 5) El usuario visualiza el resultado de la selección.

Post-Condición: El usuario visualiza un listado ordenado y agrupado de los grupos de noticias del Grupo Salesiano de Comunicación, los cuales son presentados en la sección Índice dentro del aplicativo móvil EdiBosco Noticias.

Actor beneficiado: Usuario.

Para el requerimiento funcional Permitir al usuario conocer en que sección del aplicativo móvil se encuentra, se ha establecido lo siguiente:

Diagrama de caso de uso Permitir al usuario conocer en que sección del aplicativo móvil se encuentra.

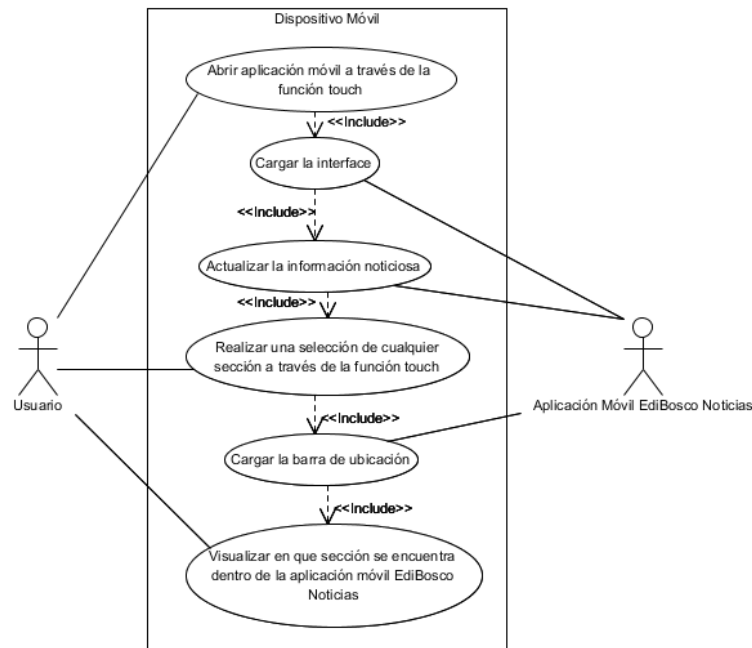


Figura 9

Elaborado por: Edwin Ceballos / Francisco Miño

Descripción de caso de uso: Permitir al usuario conocer en que sección del aplicativo móvil se encuentra.

Actores que intervienen: Usuario, Aplicación móvil.

El usuario, al abrir el aplicativo móvil, a través de un touch seleccionará una sección y visualizará en la barra de ubicación donde se encuentra dentro de la aplicación móvil.

Pre-Condición: El usuario debe seleccionar una sección a través de un touch para conocer su ubicación dentro del aplicativo móvil.

Acciones

- 1) El usuario abre el aplicativo EdiBosco Noticias a través de la función touch.
- 2) El usuario realiza una selección de cualquier sección.
- 3) El usuario visualiza en que sección se encuentra dentro del aplicativo móvil.

Post-Condición: El usuario visualiza en la barra de ubicación en que sección se

encuentra.

Actor beneficiado: Usuario.

Para el requerimiento funcional Brindar al usuario un conjunto de funciones que permitan visualizar, compartir, ordenar y refrescar las noticias, se ha establecido lo siguiente:

Diagrama de caso de uso Brindar al usuario un conjunto de funciones que permitan visualizar, compartir, ordenar y refrescar las noticias.

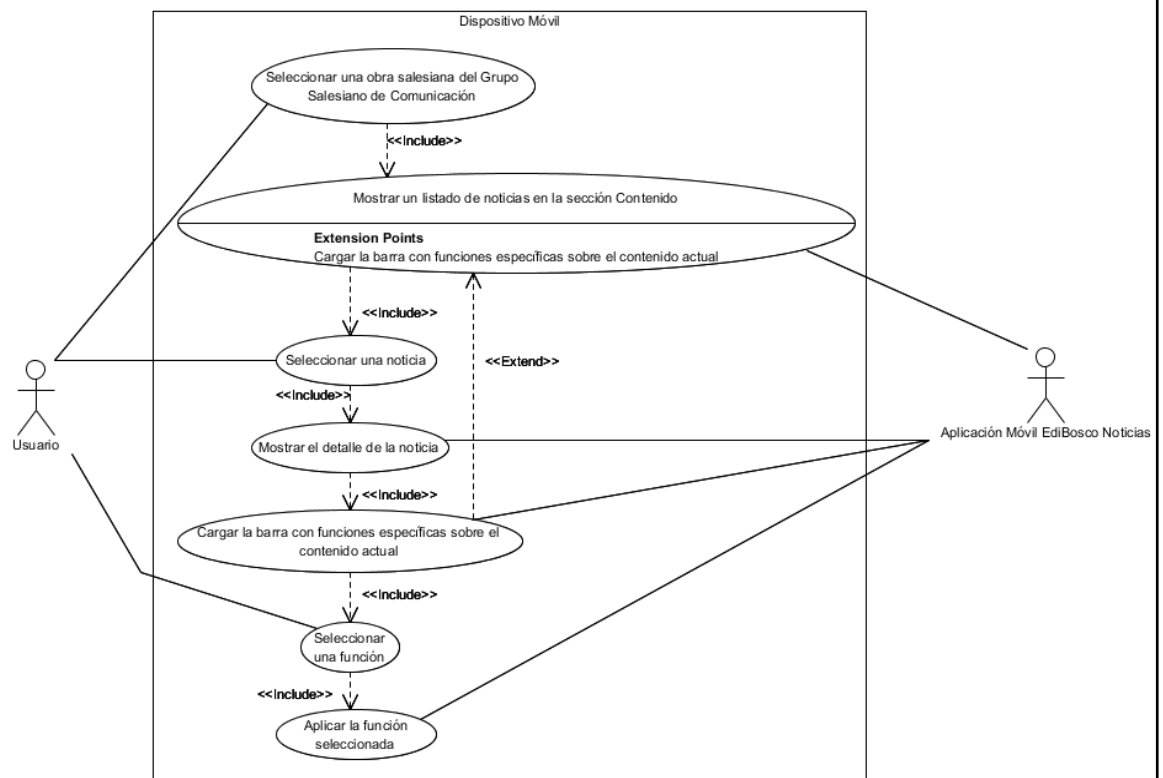


Figura 10

Elaborado por: Edwin Ceballos / Francisco Miño

Descripción de caso de uso: Brindar al usuario un conjunto de funciones que permitan visualizar, compartir, ordenar y refrescar las noticias.

Actores que intervienen: Usuario, Aplicación móvil.

El usuario al seleccionar un grupo de noticias elegirá una noticia y seleccionará una función sobre el contenido de la misma.

Pre-Condición: El usuario debe seleccionar un grupo de noticias de la sección

Índice (menú principal) a través de un touch para elegir una obra salesiana donde se despliega un listado de noticias.

Acciones

- 1) El usuario selecciona un grupo de noticias del Grupo Salesiano de Comunicación.
- 2) El usuario selecciona una noticia.
- 3) El usuario aplica una función sobre el contenido de la noticia.

Post-Condición: El usuario selecciona una función sobre el contenido de las noticias.

Actor beneficiado: Usuario.

Para el requerimiento funcional Mostrar al usuario los contenidos de las noticias de manera ordenada, se ha determinado lo siguiente:

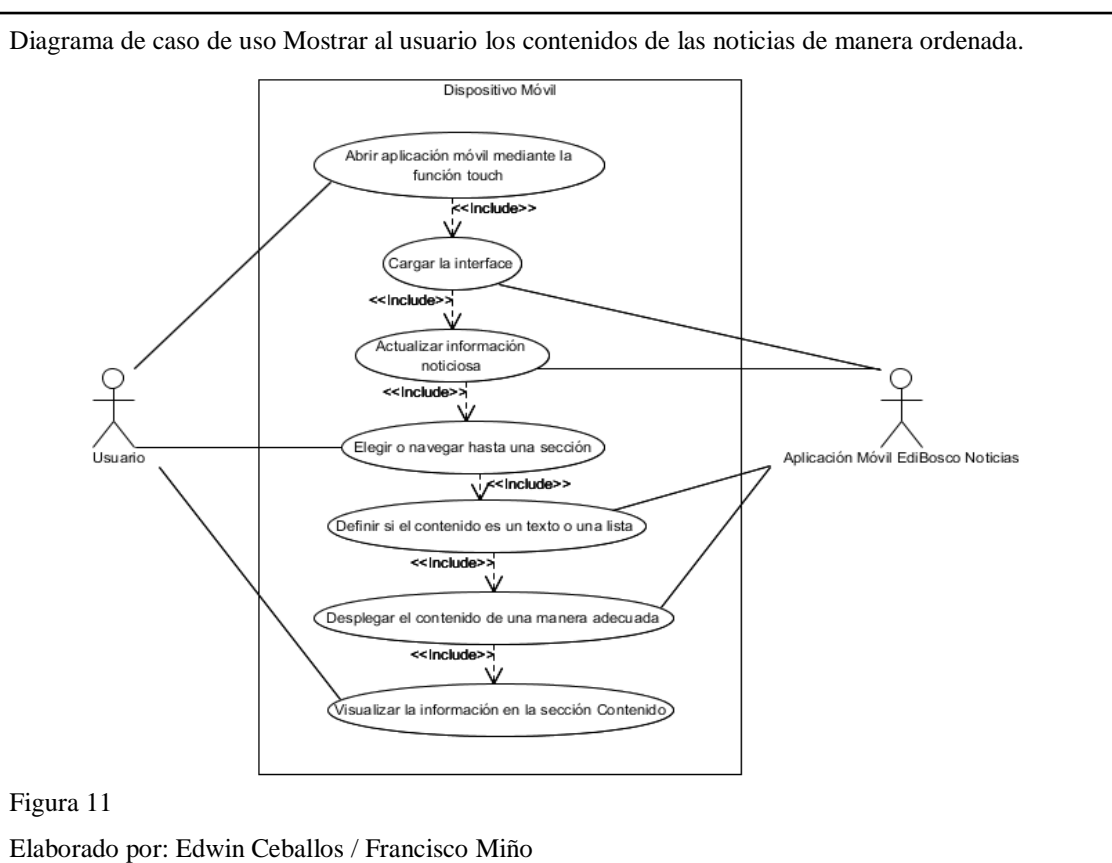


Figura 11

Elaborado por: Edwin Ceballos / Francisco Miño

Descripción de caso de uso: Mostrar al usuario los contenidos de las noticias de manera ordenada.

Actores que intervienen: Usuario, Aplicación móvil.

El usuario al seleccionar un grupo de noticias visualizará la información noticiosa en la sección Contenido.

Pre-Condición: El usuario debe seleccionar un grupo de noticias de la sección Índice (menú principal) a través de un touch y elegir una obra salesiana.

Acciones

- 1) El usuario ingresa a aplicativo EdiBosco Noticias.
- 2) El usuario escoge la sección de noticias de una obra salesiana.
- 3) El usuario visualiza la información noticiosa en la sección Contenido.

Post-Condición: El usuario visualiza la información noticiosa en la sección Contenido.

Actor beneficiado: Usuario.

Para el requerimiento funcional Brindar al usuario un listado de configuraciones aplicables a la información noticiosa, se ha establecido lo siguiente:

Diagrama de caso de uso Brindar al usuario un listado de configuraciones aplicables a la información noticiosa.



Figura 12

Elaborado por: Edwin Ceballos / Francisco Miño

Descripción de caso de uso: Brindar al usuario un listado de configuraciones aplicables a la información noticiosa.

Actores que intervienen: Usuario, Aplicación móvil.

El usuario seleccionará la opción Configuración para activar o desactivar las configuraciones disponibles y visualizará los cambios aplicados a las noticias.

Pre-Condición: El usuario debe seleccionar un grupo de noticias de la sección Índice (menú principal) a través de un touch para elegir una obra salesiana donde se despliega un listado de noticias.

Acciones

- 1) El usuario ingresa al aplicativo EdiBosco Noticias.
- 2) El usuario escoge el ícono Configuración.
- 3) El usuario activa o desactiva las configuraciones disponibles.
- 4) El usuario visualiza los cambios realizados.

Post-Condición: El usuario visualiza los cambios realizados sobre la información noticiosa.

Actor beneficiado: Usuario.

Para el requerimiento funcional Brindar al usuario accesos directos para visualización e ingreso de nuevas noticias, se ha establecido lo siguiente:

Diagrama de caso de uso Brindar al usuario accesos directos para visualización e ingreso de nuevas noticias.

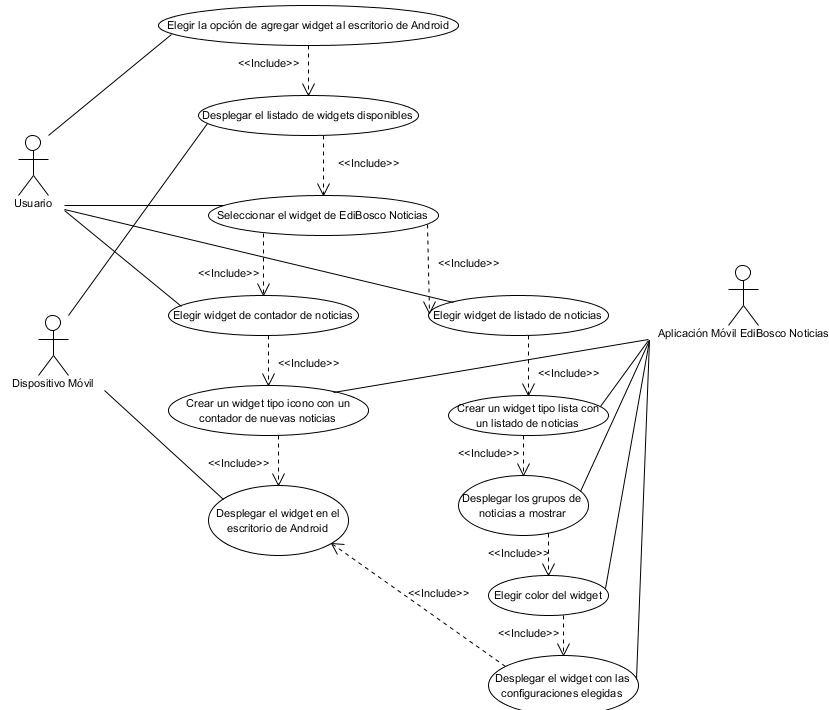


Figura 13

Elaborado por: Edwin Ceballos / Francisco Miño

Descripción de caso de uso: Brindar al usuario accesos directos para visualización e ingreso de nuevas noticias.

Actores que intervienen: Usuario, Dispositivo móvil, Aplicación móvil.

El usuario elegirá la opción de agregar el widget de la aplicación móvil al escritorio de Android y visualizará las nuevas noticias existentes.

Pre-Condición: El usuario debe agregar el widget de EdiBosco Noticias al escritorio de Android.

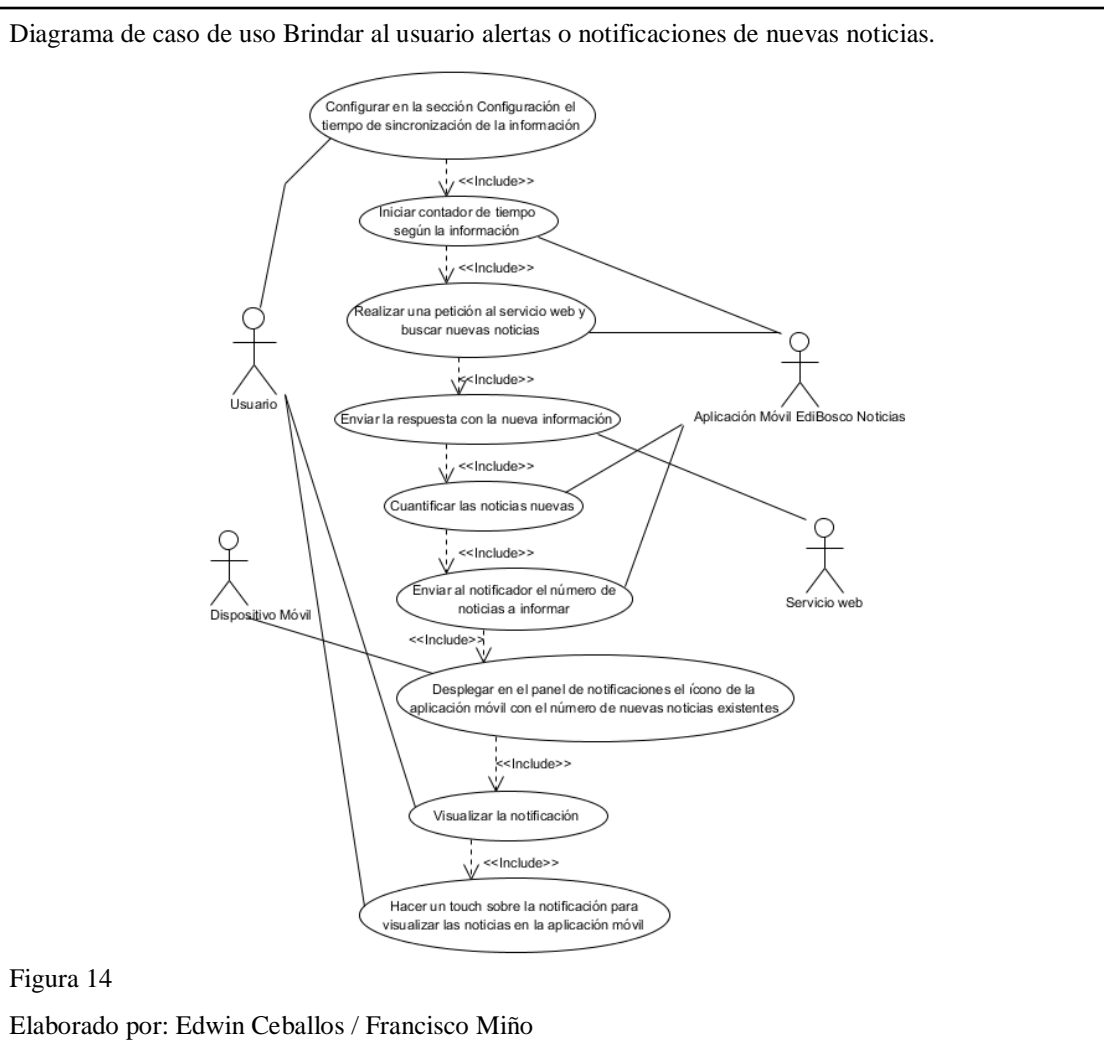
Acciones

- 1) El usuario elige la opción de agregar widget al escritorio de Android.
- 2) El usuario selecciona el widget EdiBosco Noticias.
- 3) El usuario elige el widget ya sea de contador o listado de noticias.

Post-Condición: El usuario elige el widget de contador o de listado de noticias para un acceso más rápido al aplicativo móvil.

Actor beneficiado: Usuario.

Para el requerimiento funcional Brindar al usuario alertas o notificaciones de nuevas noticias, se ha establecido lo siguiente:



Descripción de caso de uso: Brindar al usuario alertas o notificaciones de nuevas noticias.

Actores que intervienen: Usuario, Dispositivo móvil, Aplicación móvil, Servicio web.

El usuario configurará el tiempo de sincronización de la información noticiosa y visualizará en el panel de notificaciones del dispositivo móvil el número de nuevas noticias existentes.

Pre-Condición: El usuario debe configurar el tiempo de sincronización de las noticias.

Acciones

- 1) El usuario configura el tiempo de sincronización de la información noticiosa.
- 2) El usuario visualiza la notificación.
- 3) El usuario selecciona la notificación.

Post-Condición: El usuario visualiza en el panel de notificaciones del dispositivo móvil el número de nuevas noticias existentes.

Actor beneficiado: Usuario.

- Tareas del webmaster

A continuación se esquematizan las tareas que realiza el webmaster al momento de administrar las noticias:

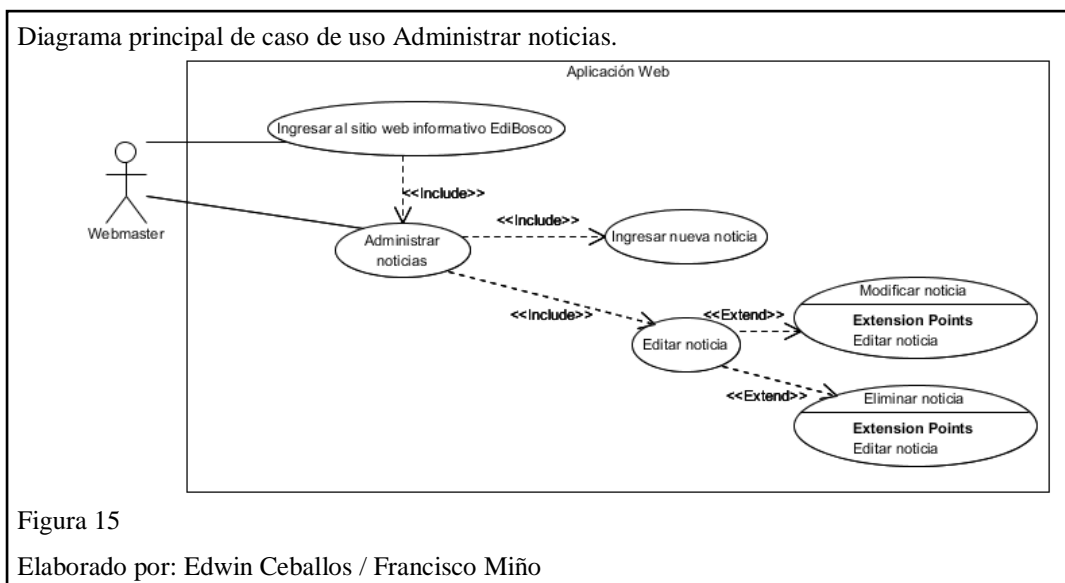


Figura 15

Elaborado por: Edwin Ceballos / Francisco Miño

El webmaster consultará la noticia que se encuentra registrada en el servicio web de la Editorial Don Bosco, en el momento que el webmaster administra la noticia, solicitará la búsqueda de la información de la noticia correspondiente al grupo seleccionado previamente y posteriormente mostrará los resultados al usuario cuando éste ingrese al aplicativo EdiBosco Noticias.

Descripción tarea: Ingresar nueva noticia.

Actores: Webmaster

El webmaster ingresará la información correspondiente de una nueva noticia, luego pedirá al sistema que valide la información (según las reglas del negocio establecidas por la empresa), así como verificará la integridad de los datos, después la noticia quedará registrada y a disposición para que ésta sea visualizada por el usuario.

Pre-Condición: El usuario a realizar esta tarea debe ser webmaster del sitio web informativo de la editorial, del servicio web y de la base de datos del sistema.

Acciones

1) El webmaster ingresa los datos de una nueva noticia a ser registrada en el sistema, los datos son guardados en el servicio web y los resultados son vistos por el webmaster.

Post-Condición: El webmaster visualiza cambios realizados.

Descripción tarea: Editar noticia.

Actores: Webmaster

El webmaster seleccionará y editará los datos de una noticia en particular que se encuentra registrada en el servicio web, luego realizará una consulta y mostrará los resultados al usuario.

Pre-Condición: El usuario a realizar esta tarea debe ser webmaster del sitio web informativo de la editorial, del servicio web y de la base de datos del sistema.

Acciones

1) El webmaster selecciona la noticia a ser buscada en el sitio web informativo, la información de la noticia seleccionada es desplegada en la pantalla del sistema, realizando la acción correspondiente.

Post-Condición: El webmaster visualiza cambios realizados.

Descripción tarea: Modificar noticia.

Actores: Webmaster

El webmaster modificará los datos de alguna noticia en particular, para lo cual es necesario la realización del caso de uso Editar Noticia (según las reglas del negocio establecidas por la empresa), luego el sistema validará la noticia, así como verificará la integridad de los datos, si la modificación de los datos de la noticia es viable, el sistema guardará los cambios realizados en dicha noticia.

Pre-Condición: El usuario a realizar esta tarea debe ser webmaster del sitio web informativo de la editorial, del servicio web y de la base de datos del sistema.

Acciones

1) El webmaster modifica los datos permitidos para realizar esta acción, de la noticia previamente editada, los cambios realizados en la información de la noticia son reflejados en el servicio web y los resultados de la modificación de los datos se muestran en la pantalla del sistema.

Post-Condición: El webmaster visualiza cambios realizados.

Descripción tarea: Eliminar noticia.

Actores: Webmaster

El webmaster eliminará los datos de una noticia en particular para lo cual es necesario la realización del caso de uso Editar noticia, dicha noticia es eliminada y como respuesta pedirá al sistema que confirme la eliminación de la noticia, si la eliminación de los datos es aceptada, se procederá con la eliminación de dicha noticia en el servicio web.

Pre-Condición: El usuario a realizar esta tarea debe ser webmaster del sitio web informativo de la editorial, del servicio web y de la base de datos del sistema.

Acciones

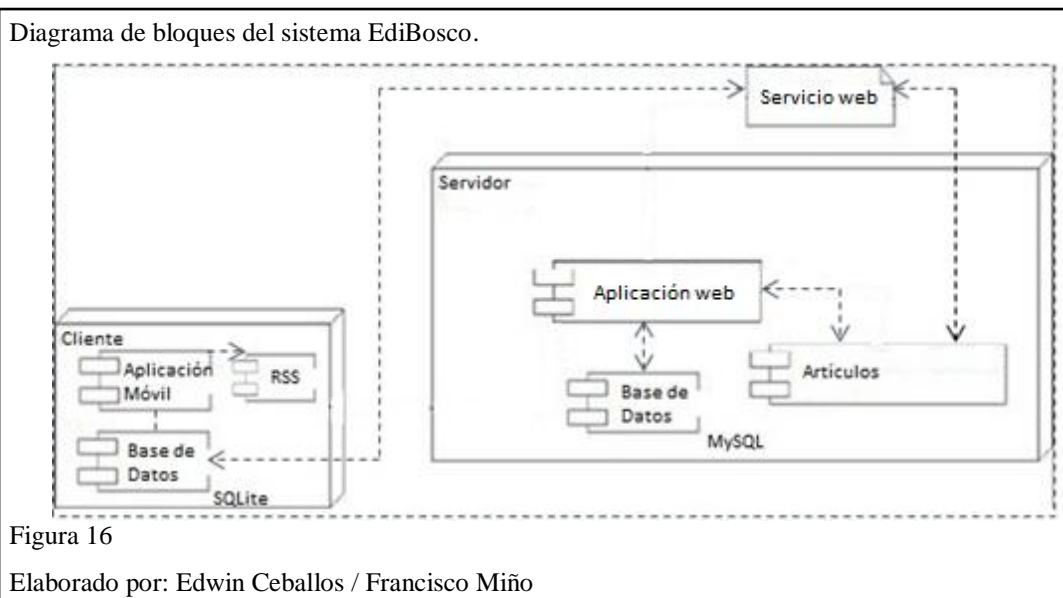
1) El webmaster elimina los datos de la noticia previamente editada, los cambios realizados son reflejados en el servicio web, eliminando a la noticia y los resultados

se muestran en la pantalla del sistema.

Post-Condición: El webmaster visualiza cambios realizados.

3.3.2 Diagrama de bloques

El diagrama de bloques describe el funcionamiento interno del sistema mediante bloques y sus relaciones. A continuación se muestra el siguiente diagrama de bloques:

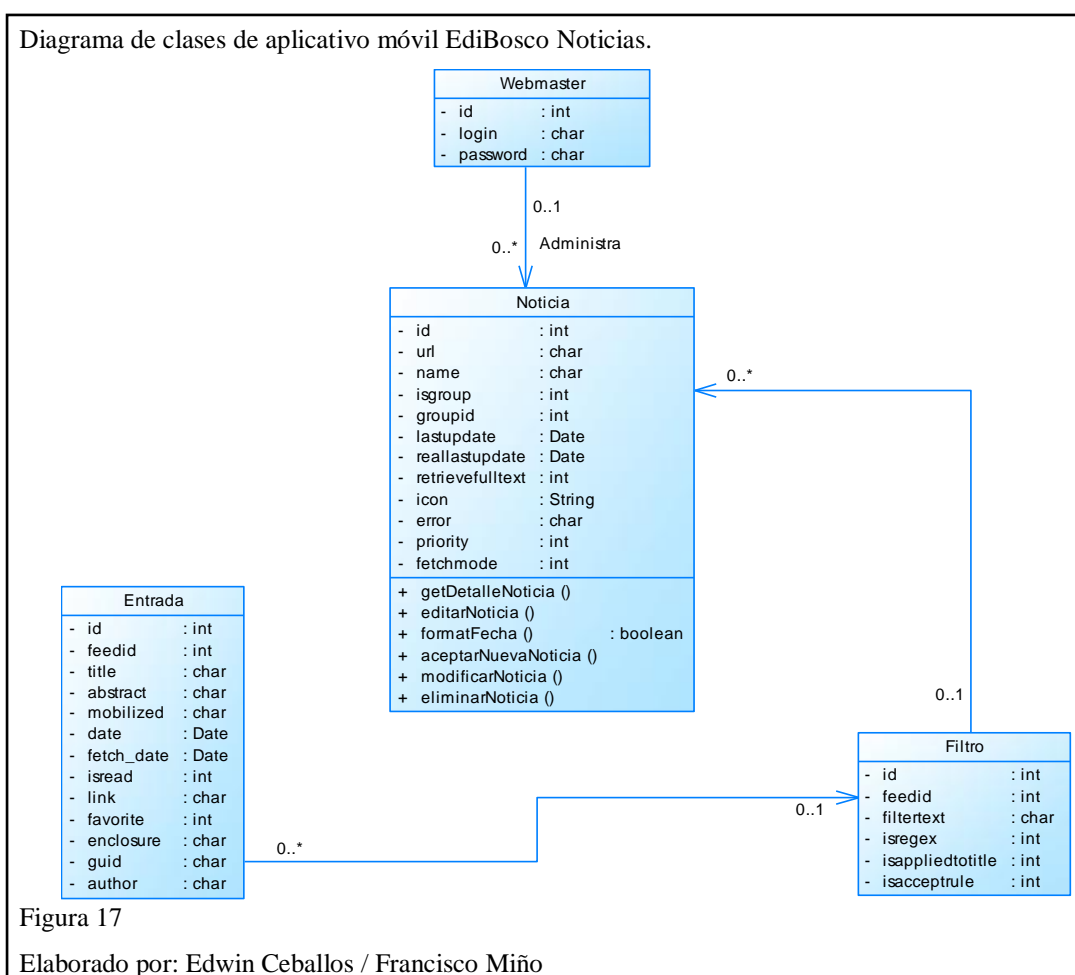


Se muestran los bloques del sistema:

- El bloque Cliente representado por el aplicativo móvil mostrará lo que será la interfaz, significa que la aplicación móvil recepta la información de la aplicación web a través del servicio web y la muestra al usuario como noticia, la cual en su estructura es un archivo RSS.
- El bloque Servidor, contiene la implementación de los procesos del sistema, significa que la aplicación web muestra los artículos (ítems o noticias) almacenados en su base de datos en el sitio web y a través del servicio web envía la información a la aplicación móvil.

3.3.3 Diagrama de clases

El diagrama de clases permite visualizar las relaciones entre las clases que involucran el aplicativo móvil, (4 clases principales con sus respectivos campos), donde la clase Webmaster se encarga de administrar las noticias, las mismas que pasan por la clase Filtro, para luego mostrar una nueva noticia (representada por la clase Entrada) al usuario. A continuación se muestra el siguiente diagrama de clases:



El diagrama de base de datos define la estructura del sistema, es decir muestra las entidades relevantes para el sistema de información (4 entidades principales), así como sus interrelaciones y propiedades. Para mayor detalle se documentó el respectivo diccionario de datos (Ver Anexo 2). A continuación se muestra el siguiente diagrama de base de datos:

Diagrama de base de datos de aplicativo móvil EdiBosco Noticias.

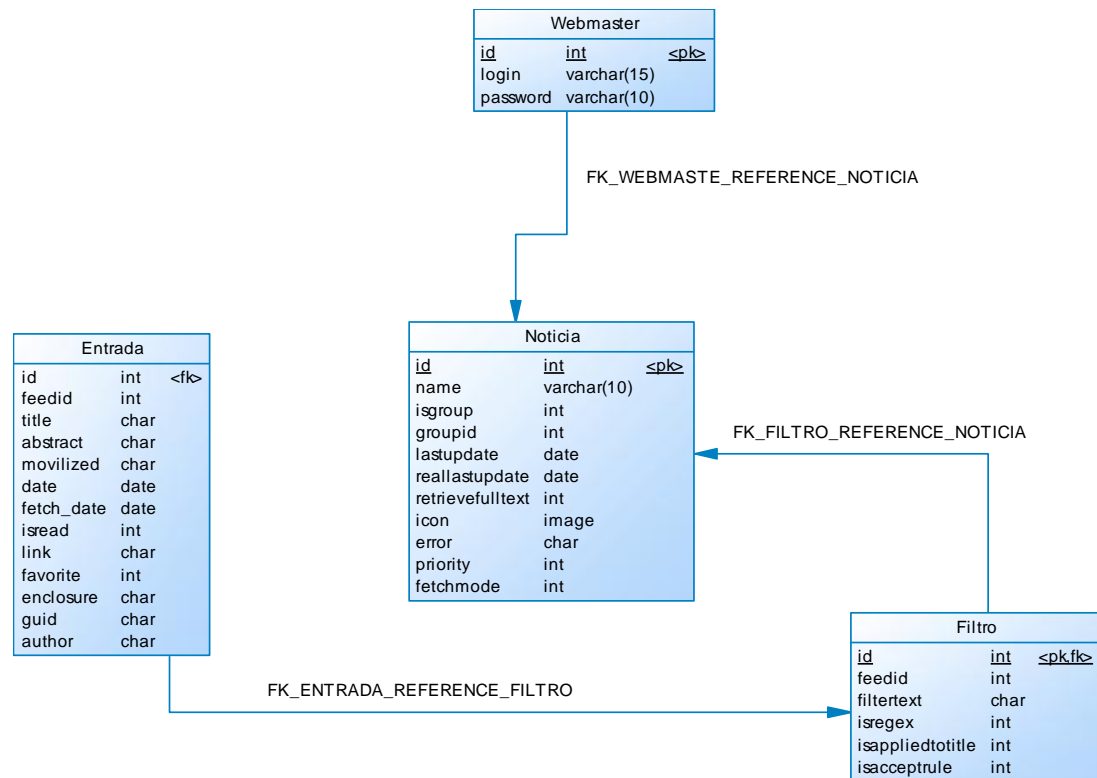


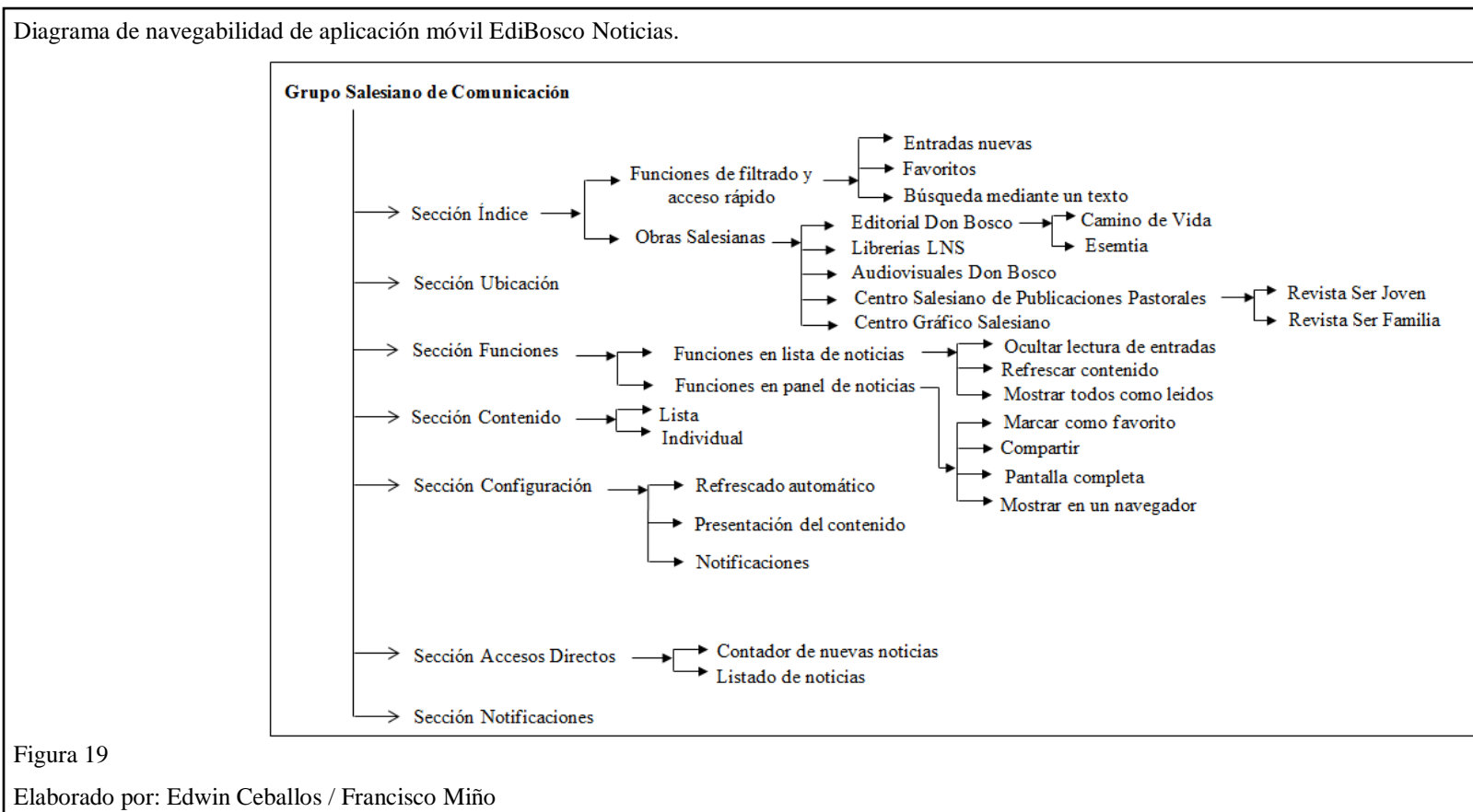
Figura 18

Elaborado por: Edwin Ceballos / Francisco Miño

3.3.4 Diagrama de navegabilidad

El diagrama de navegabilidad sirve para mostrar la interactividad que existe entre las pantallas especificadas (secciones) de la aplicación móvil.

A continuación se muestra el siguiente diagrama de navegabilidad:



3.3.5 Diagrama de interfaz de usuario

De acuerdo a lo mencionado en el punto 3.1.1.1, a continuación se presenta el siguiente diagrama de interfaz de usuario:

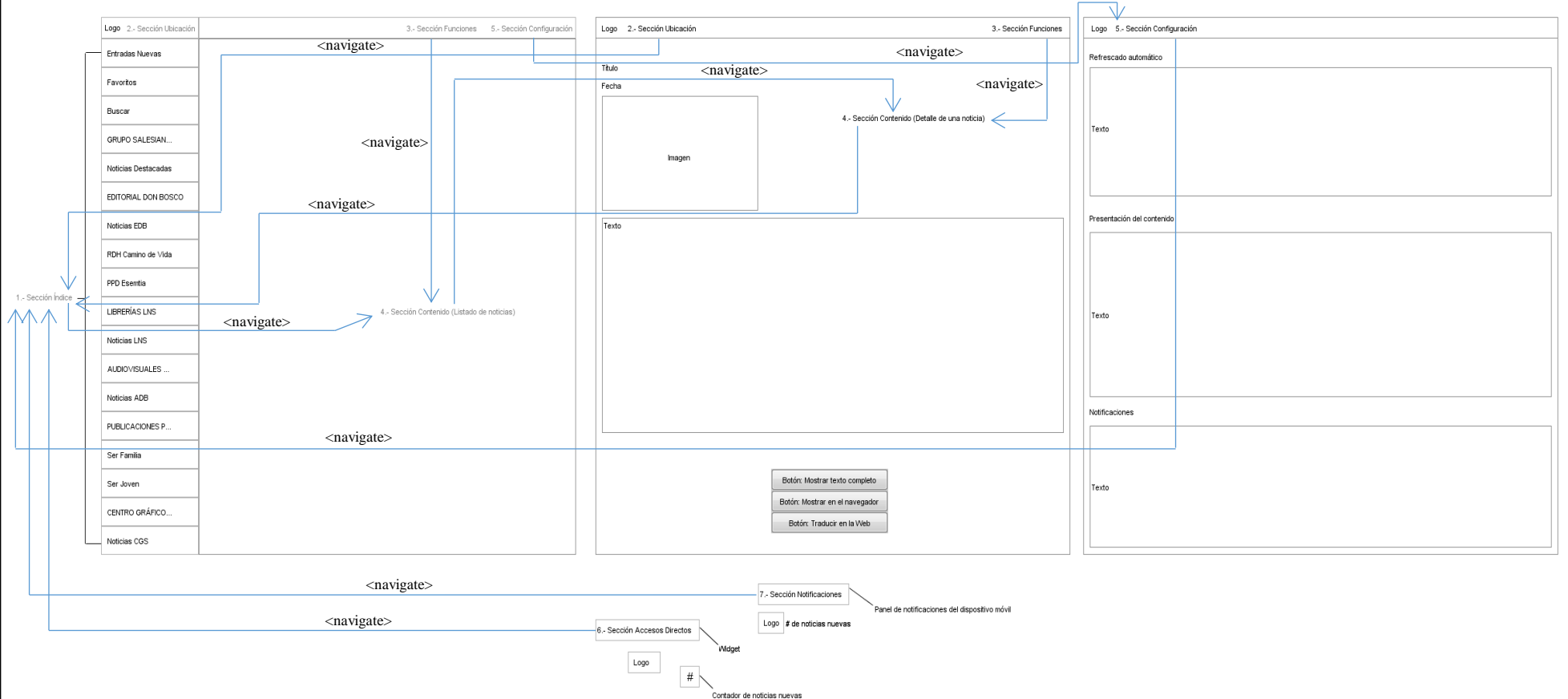
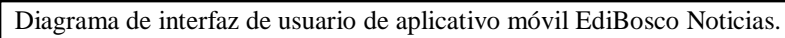


Figura 20

Elaborado por: Edwin Ceballos / Francisco Miño

Se esquematizan las principales interfaces de la aplicación móvil, representada por fechas, la interconectividad que existe entre secciones.

CAPÍTULO 4

CONSTRUCCIÓN

En este capítulo se indican los estándares de programación, se describe la arquitectura utilizada para el desarrollo de la aplicación móvil y los pasos llevados a cabo para la construcción del aplicativo móvil, servicio web e interfaces de usuario.

4.1 Estándares de programación

Los estándares de programación contienen directivas para el desarrollo de software, con el objetivo de facilitar la estandarización y el mantenimiento del código fuente de la aplicación móvil desarrollada.

4.1.1 Buenas prácticas para manejo de tablas

Se deben tomar en cuenta los siguientes lineamientos para el manejo de tablas:

- Usar nombres claros y bien definidos para las tablas, evitando la ambigüedad (ejemplo: Noticia, Entrada, etc.).
- Usar nombres en singular para las tablas (Noticia en lugar de Noticias). La tabla representa una colección de entidades, pero no es necesario usar nombres en plural.
- No incluir espacios en los nombres de las tablas.
- Usar prefijos claros. A continuación se detallan los controles con sus respectivos prefijos utilizados en tablas.

Tabla 10

Controles y prefijos utilizados en tablas.

Control	Prefijo
Label	Lbl
TextBox	Txt
DataGrid	Dtg
Border	Bor
Grid	Gr
Button	Btn
HyperLinkButton	Hlb
ListBox	Lst
Calendar	Cal
CheckBox	Chk
RadioButton	Rdo

Control	Prefijo
Image	Img
Canvas	Cnv
PasswordBox	Psb
Rectangle	Rect
Slider	Sld
ComboBox	Cmb

Nota: Detalle general

Elaborado por: Edwin Ceballos / Francisco Miño

- Usar tipo de datos enteros como identificadores para todas las tablas.
- Todos los programas inician con la palabra reservada *class*.
- El nombre del archivo del código fuente debe ser igual al nombre de la clase y terminan con la extensión *.java*.

4.1.2 Buenas prácticas para manejo de instrucciones

Se deben tomar en cuenta los siguientes lineamientos para el manejo de instrucciones:

- Evitar las líneas de más de 80 caracteres, ya que no son manejadas bien por muchas terminales y herramientas.
- Cuando una expresión no entre en una línea, habrá que romperla de acuerdo con estos principios:
 - Romper después de una coma.
 - Romper antes de un operador.
 - Preferir roturas de alto nivel (más a la derecha que el "padre") que de bajo nivel (más a la izquierda que el "padre").
 - Alinear la nueva línea con el comienzo de la expresión al mismo nivel de la línea anterior.
- La primera línea no-comentario de los archivos fuente Java es la sentencia *package*. Después de esta, pueden seguir varias sentencias *import*.

4.1.3 Buenas prácticas para manejo de comentarios

Se deben tomar en cuenta los siguientes lineamientos para el manejo de comentarios:

- Los comentarios de documentación (conocidos como "doc comments") existen sólo en Java, y se limitan por `/**...*/`, estos se pueden exportar a archivos HTML con la herramienta *javadoc*.
- Comentar áreas de código que se consideren complejas en lenguaje español, debido a que sirve para describir de qué trata cierto código, para orientar a otros desarrolladores sobre que hace el código o para comentar código que no se quiere borrar. Los comentarios deben contener sólo información que es relevante para la lectura y entendimiento de la aplicación móvil.

4.1.4 Buena práctica para manejo de declaraciones

Se debe tomar en cuenta el siguiente lineamiento para el manejo de declaraciones:

- Se recomienda una declaración por línea, ya que facilita los comentarios.

4.1.5 Regla de nomenclatura en variables y métodos

Para nombres compuestos de varias palabras se capitaliza la primera letra de cada palabra con excepción de la primera palabra donde todas son minúsculas, esta regla se conoce como Camel, la cual, según su autor menciona que la primera letra del identificador está en minúscula y la primera letra de las siguientes palabras concatenadas en mayúsculas (Fosch & Arellano, 2006). Ejemplo: *whiteColor*.

Para las variables se usa nombres cortos y descriptivos en sí mismos. Ejemplos: *title* y *name*. La elección del nombre de las variables debe ser nemotécnica, es decir, pensado para que un lector casual al verla comprenda su uso, como lo indica el siguiente fragmento de código fuente.

```
import android.app.NotificationManager;
```

```
import android.content.Context;
```

```
public final class Constants {

    public static final NotificationManager NOTIF_MGR =
(NotificationManager)
MainApplication.getContext().getSystemService(Context.NOTIFICATION);

    public static final String INTENT_FROM_WIDGET = "fromWidget";

    public static String FEED_ID = "feedid";
```

Se deben evitar las variables de una sola letra, excepto en variables temporales de corto uso. Nombres comunes para este tipo de variables son: *i, j, k, m* y *n* para enteros; *c, d*, y *e* para caracteres.

Para los métodos se utiliza verbos. Ejemplo: *onStartCommand()*.

4.1.6 Regla de nomenclatura en clases, propiedades, componentes y controles

Para nombres compuestos de varias palabras se capitaliza la primera letra de cada palabra, esta regla se conoce como Pascal, la cual, según su autor menciona que la primera letra del identificador y la primera letra de las siguientes palabras concatenadas están en mayúsculas (Fosch & Arellano, 2006). Ejemplo: *WhiteColor*.

Para las clases se usa acrónimos o abreviaturas para reemplazar palabras extensas de más de diez letras (a no ser que la abreviatura sea ampliamente conocida, como URL o HTML). Ejemplo: GSC (Grupo Salesiano de Comunicación). De igual manera no debe existir ningún espacio en blanco entre el nombre de un método y el paréntesis "(" que abre su lista de parámetros, también la llave de apertura "{" aparece al final de la misma línea de la sentencia declaración y la llave de cierre "}" se debe ajustar a su sentencia de apertura correspondiente, excepto cuando no existen sentencias entre ambas, que debe aparecer inmediatamente después de la de apertura "{" como se muestra en el siguiente fragmento de código fuente.

```
public class MainApplication extends Application {  
    private static Context context;  
    @Override  
    public void onCreate() {  
        super.onCreate();  
        context = getApplicationContext();  
        PrefUtils.putBoolean(PrefUtils.IS_REFRESHING, false);  
    }  
    public static Context getContext() {  
        return context;  
    }  
}
```

Para las propiedades, componentes y controles se utilizan sustantivos. Ejemplos: *CheckBox*, *TextView* y *ImageView*, como se indican en los siguientes fragmentos de código fuente.

```
package net.edwc.notibosco.adapter;

import android.content.ContentResolver;

    public class EntriesCursorAdapter extends ResourceCursorAdapter {
        private static class ViewHolder {

            public TextView titleTextView;
            public TextView dateTextView;
            public ImageView starImgView;
            public CheckBox isReadCb;

        }

        private int mTitlePos, mDatePos, mIsReadPos, mIdPos, mFeedIconPos,
mFeedNamePos;

        private final Uri mUri;
        private final Boolean mShowFeedInfo;

        private final Vector<Long> mMarkedAsReadEntries = new
Vector<Long>();

        private final Vector<Long> mMarkedAsUnreadEntries = new
Vector<Long>();

        private final Vector<Long> mFavoriteEntries = new Vector<Long>();
        private final Vector<Long> mNotFavoriteEntries = new Vector<Long>();

        public EntriesCursorAdapter(Context context, Uri uri, Cursor cursor,
Boolean showFeedInfo) {

            super(context, R.layout.item_entry_list, cursor, 0);

            mUri = uri;

            mShowFeedInfo = showFeedInfo;

        }

        @Override

        public void bindView(View view, final Context context, Cursor cursor) {

            if (view.getTag() == null) {

                ViewHolder holder = new ViewHolder();
```

```

        holder.titleTextView = (TextView)
view.findViewById(android.R.id.text1);
        holder.dateTextView = (TextView)
view.findViewById(android.R.id.text2);
        holder.starImgView = (ImageView)
view.findViewById(android.R.id.icon);
        holder.isReadCb = (CheckBox)
view.findViewById(android.R.id.checkbox);
        view.setTag(holder);
    }
    public void bindView(View view, final Context context, Cursor cursor) {
        if (view.getTag() == null) {
            ViewHolder holder = new ViewHolder();
            holder.titleTextView = (TextView)
view.findViewById(android.R.id.text1);
            holder.dateTextView = (TextView)
view.findViewById(android.R.id.text2);
            holder.starImgView = (ImageView)
view.findViewById(android.R.id.icon);
            holder.isReadCb = (CheckBox)
view.findViewById(android.R.id.checkbox);
            view.setTag(holder);
        }
    }

```

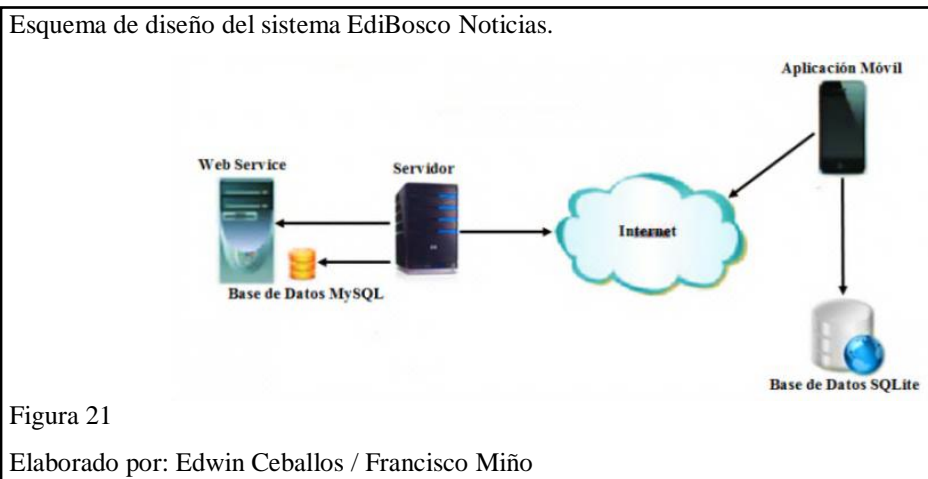
4.1.7 Regla de recuperación de errores

Se debe tomar en cuenta el siguiente lineamiento para el manejo de errores:

- Asignar mensajes significativos que permitan al usuario entender claramente el problema (Ruiz & Estévez, 2012).

4.2 Arquitectura del sistema

El diseño general del software está formado por tres componentes principales: Un servicio web, una aplicación web y una aplicación móvil, las dos últimas pueden acceder al servicio web a través de Internet. El servidor que aloja el servicio web también contiene la base de datos. El servidor de aplicaciones manejará el procesamiento de la información y la lógica del negocio.



4.2.1 Funcionamiento del servicio web EdiBosco

El servicio web (detallado en la figura 22), como resultado de todas las etapas de desarrollo utilizando la metodología Mobile-D funciona así: el usuario ejecuta la aplicación desde el dispositivo móvil; ésta se conecta al servicio web del sistema y realiza una consulta al mismo a través de: la red móvil o Internet, como lo muestra el siguiente fragmento de código fuente.

```
package net.edwc.notibosco.receiver;  
  
import android.content.BroadcastReceiver;  
  
import android.content.Context;  
  
import android.content.Intent;  
  
import android.net.ConnectivityManager;  
  
import android.os.SystemClock;  
  
import net.edwc.notibosco.Constants;  
  
import net.edwc.notibosco.service.FetcherService;
```

```

import net.edwc.notibosco.service.RefreshService;

import net.edwc.notibosco.utils.PrefUtils;

    public class ConnectionChangeReceiver extends BroadcastReceiver {

        private boolean mConnection = false;

        @Override

        public void onReceive(Context context, Intent intent) {

            if (mConnection &&

intent.getBooleanExtra(ConnectivityManager.EXTRA_NO_CONNECTIVITY, false))
{

                mConnection = false;

            } else if (!mConnection &&

!intent.getBooleanExtra(ConnectivityManager.EXTRA_NO_CONNECTIVITY,
false)) {

                mConnection = true;

                if (!PrefUtils.getBoolean(PrefUtils.IS_REFRESHING, false) &&
PrefUtils.getBoolean(PrefUtils.REFRESH_ENABLED, true)) {

                    int time = 3600000;

                    try {

                        time = Math.max(60000,

Integer.parseInt(PrefUtils.getString(PrefUtils.REFRESH_INTERVAL,
RefreshService.SIXTY_MINUTES)));

                    } catch (Exception ignored) {

                    }

                    long lastRefresh = PrefUtils.getLong(PrefUtils.LAST_SCHEDULED_REFRESH, 0);

                    if (SystemClock.elapsedRealtime() - lastRefresh > time) {

                        context.startService(new Intent(context,

```

```

    FetcherService.class).setAction(FetcherService.ACTION_REFRESH_FEEDS).putExtra(Constants);

    }

    }

    }

}

```

Recibiendo el dispositivo móvil respuesta de parte del servicio web a través del mismo camino y guardando la información en la base de datos de la aplicación móvil, como lo muestra el siguiente fragmento de código fuente.

```

import android.content.Context;

public abstract class BaseLoader<D> extends AsyncTasksLoader<D> {

    private D data;

    public BaseLoader(Context context) {

        super(context);

    }

    @Override

    public void deliverResult(D data) {

        if (isReset()) {

            return;

        }

        this.data = data;

        super.deliverResult(data);

    }

    @Override

```



```

protected void OnStartLoading() {

    if (data != null) {

        deliverResult(data);

    }

    If (takeContentChanged() || data == null ) {

        forceLoad();

    }

}

@Override

protected void onStartLoading() {

    cancelLoad();

}

@Override

protected void onReset() {

    super.onReset();

```

Arquitectura del servicio web EdiBosco.

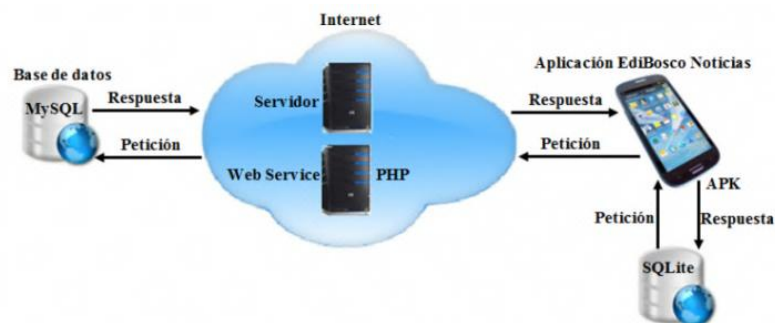


Figura 22

Elaborado por: Edwin Ceballos / Francisco Miño

En la composición del servidor funcionan en realidad tres componentes básicos:

- Una base de datos, que almacena toda la información que envían los usuarios.

- Un servicio web que atiende la petición recibida, la procesa y envía la respuesta correspondiente.
- Un servidor, donde reside y se ejecuta el servicio web, y que permanece a la espera de conexiones HTTP entrantes por medio de Internet, como lo indica la figura 21.

El servicio web es publicado en un servidor único, entonces otras aplicaciones que necesitan acceder al servicio web obtiene la información del mismo, y esto es posible debido a que un servicio web permite la integración de varios estándares, como por ejemplo Simple Access Protocol (SOAP), Web Service Definition Language (WSDL) y Universal Description Discovery and Integration (UDDI).

4.2.2 Análisis modelo vista controlador de aplicación móvil EdiBosco Noticias

Dado que el sitio web de la Editorial Don Bosco utiliza Joomla, se basa en un patrón solicitado por este programa a los desarrolladores de extensiones como una manera de estandarizar la estructura de los componentes, módulos, plugins, etc. en tres dimensiones, separando: Los datos de una aplicación (modelo), la interfaz de usuario (vista) y la lógica de control (controlador). De esta manera se tiene una arquitectura de software ordenada y limpia, que permite entender el código más rápidamente, crear extensiones con mucha facilidad, así como intervenir el código sin mayores complicaciones, como se muestra en el siguiente esquema:

Tabla 11

Modelo vista controlador de aplicación móvil EdiBosco Noticias.

Modelo	Vista	Controlador
SQLite	XML	Java
(Base de Datos)	App	(Código)
(Consultas)	(Interface)	(Consultas)
	(Consultas)	

Nota: Esquema general

Elaborado por: Edwin Ceballos / Francisco Miño

El modelo representa las estructuras de datos (capa del negocio); es decir toda la información con la que opera la aplicación; gestionar el comportamiento y los datos

del dominio; típicamente las clases del modelo contendrán funciones que ayudan a devolver, insertar y actualizar información de su base de datos, independientemente de la base de datos a utilizar o posibles cambios. Encapsular el modelo de una aplicación en componentes facilita la depuración, mejora la calidad y favorece la reutilización de código, y los cambios que el sistema pudiera sufrir a futuro.

La vista es la información que se presenta al usuario. La vista gestiona la presentación de la información de la aplicación; todo lo relativo a la interfaz de usuario, los datos necesarios para que el usuario pueda seguir interactuando con la aplicación. Una vista será normalmente una página web, en este caso es el sitio web informativo de la Editorial Don Bosco y la interfaz del aplicativo móvil EdiBosco Noticias, La vista representa la lógica de presentación de la aplicación. Los componentes de la vista extraen el estado actual del sistema del modelo y proporcionan la interfaz de usuario para el protocolo que se está usando.

El controlador es el indicado a responder a los eventos invocados desde la vista; y a su vez llamar a los métodos correspondientes el modelo para procesar y producir las respuestas necesarias con los datos adecuados para la interacción de la aplicación. El controlador sirve como un intermediario entre el modelo, la vista y cualquier otro recurso necesario para procesar la solicitud HTTP y generar una página web. El controlador es responsable de recibir los eventos, determinar el procesador del evento, invocar al procesador y finalmente provocar la generación de la vista apropiada; y debe realizar las siguientes tareas:

- Control de la seguridad.
- Identificación de eventos.
- Preparar el modelo.
- Procesar el evento.
- Manejar los errores.
- Provocar la generación de la respuesta.

4.3 Construcción de las soluciones de software

Para obtener la información que se encuentra en la base de datos que utiliza el sitio web de Editorial Don Bosco, se necesita de un servicio web para que la aplicación

móvil consume los datos de dicho sitio web informativo, es por eso que se procedió a construir lo siguiente:

4.3.1 Construcción de la aplicación móvil

Para el desarrollo de la aplicación móvil se tomó en cuenta los siguientes aspectos:

- Utilizar Eclipse Juno 4.2 (detallado en el punto 2.3.2.1), como herramienta principal de desarrollo, debido a que resulta el entorno de desarrollo más recomendable para Android, es libre y además es soportado por Google (utilizado por los desarrolladores de Google para crear Android).
- Usar SDK Android (detallado en el punto 2.3.2.2), como herramienta de codificación de la aplicación móvil, debido a que presenta un conjunto de librerías desarrolladas por Google para construir, compilar y depurar aplicaciones para el sistema operativo Android.
- Utilizar base de datos SQLite (detallada en el punto 2.4.1), para el almacenamiento de información suministrada en el aplicativo, debido a que no requiere configuración y no tiene un servidor de base de datos ejecutándose en un proceso separado.

Para comenzar con la construcción de la aplicación móvil EdiBosco Noticias, se necesitó instalar y configurar las siguientes herramientas:

- Java Development Kit (JDK) (Ver Anexo 3).
- Android Standard Development Kit (SDK) (Ver Anexo 4).
- Eclipse Juno versión 4.2 (Ver Anexo 5).
- Plugin Android Developer Tools (ADT) para Eclipse (Ver Anexo 6).
- Uno o más dispositivos móviles para ejecutar la aplicación.

Para la realización de este trabajo de grado se necesitó tener conocimientos en Java y programación orientada a objetos.

Una vez instaladas y configuradas las herramientas mencionadas en el punto 4.3.1, se procedió a la construcción del aplicativo EdiBosco Noticias (Ver Anexo 7).

La aplicación móvil internamente se compiló de acuerdo a lo mencionado en el punto 2.1.2, donde se detalla el proceso paso a paso.

4.3.2 Construcción del servicio web

El servicio web representa un componente de software de gran utilidad ya que permite la comunicación entre el aplicativo móvil y la aplicación web de la cual se va a recoger la información, para esto el servicio web está estructurado de manera que la información recolectada podrá ser tratada para una visualización adecuada en el dispositivo móvil sin que esto represente la alteración del formato original de la misma.

Para alcanzar el objetivo de obtener y tratar la información de la aplicación web, el servicio web maneja la siguiente estructura básica:

- Inicialmente se cuenta con un archivo llamado *conexión.php*, el cual crea la conexión a la base de datos y comprueba la existencia de información.
- Se crea el paquete de datos, con la sentencia *header('Content-type: application/json');*
- Se inicializa los datos de conexión con las siguientes sentencias.

```
$server = "localhost";  
$username = "inscompr_useredb";  
$password = "useredb";  
$database = "inscompr_edibosco";
```

- Se crea un objeto de conexión a la base de datos con el siguiente fragmento de código fuente.

```
$con = mysql_connect($server, $username, $password) or die ("Could not connect:  
" . mysql_error());  
mysql_select_db($database, $con);
```

- Se ejecuta la consulta y se genera un resultado con las siguientes sentencias.

```
$sql = "SELECT * FROM edbh_content ORDER BY id ASC";  
$result = mysql_query($sql) or die ("Query error: " . mysql_error());  
$records = array();
```

```
while($row = mysql_fetch_assoc($result)) {
    $records[] = $row;
}
```

- Se cierra la conexión con la siguiente sentencia *mysql_close(\$con)*;

- Se envía el paquete de datos procesados de la información con el siguiente fragmento de código fuente.

```
echo $_GET['jsoncallback'] . '(' . json_encode($records) . ');';
returns $records[];
```

- Se recibe el paquete de información con la sentencia *header('Content-Type: text/xml')*;

- Se crea formato para RSS 2.0, basada en la experiencia obtenida en el manejo de estos datos se recomienda usar formatos de tipo RSS versión 2.0 ya que estos permiten manejar diferentes etiquetas XML, indicado en el siguiente fragmento de código fuente.

```
echo '<?xml version="1.0" encoding="iso-8859-1" ?>';
echo '<rss version="2.0">
<channel>
<title>Ultimas Noticias</title>
<link>http://www.edibosco.com/</link>
<description>Ultimas noticias expuestas en web</description>
<language>es-ES</language>';
```

- Se carga la información en formato lista en los paquetes XML (RSS) definidos anteriormente con el siguiente fragmento de código fuente.

```
while ($fila = mysql_fetch_array($resultado)) {
    echo '<item>
<titulo>' . $fila["title"] . '</titulo>
<fecha_crea>' . $fila["created"] . '</fecha_crea>
<link>' . $fila["title"] . '</link>
<desc><![CDATA[' . $fila["introtxt"] . ']]></desc>
```

```

    <id>' . $fila["id"] . '</id>
  </item>';
}

```

Esto quiere decir que si se crea un ítem XML de una noticia se puede agregar varios detalles referentes a la misma.

- Y finalmente el servicio web presenta el documento procesado para ser leído por la aplicación móvil y como resultado se obtiene la noticia o ítem, indicado en el siguiente fragmento de código fuente.

```

<rss version="2.0">
  <channel>
    <title>Ultimas Noticias</title>
    <link>http://www.edibosco.com/</link>
    <description>Ultimas noticias expuestas en web</description>
    <language>es-ES</language>
    <item>
      <titulo>Página en construcción</titulo>
      <fecha_crea>2011-01-01 00:00:01</fecha_crea>
      <link>Página en construcción</link>
      <desc>
        <![CDATA[
          <p>El <a href="http://www.webempresa.com/creacion-web-joomla/pack-joomlawebempresa.html" title="Pack Joomla con extensiones imprescindibles">Pack Webempresa Joomla 2.5</a> ha sido correctamente
          <strong>instalado</strong>!</p>
          <p>Joomla! hace que construir su sitio así como lo desea y mantenerlo actualizado sea fácil.</p> <p>Joomla! es una plataforma flexible y poderosa, tanto si se está construyendo un pequeño sitio, como si se trata de un sitio enorme, con cientos de miles de visitantes. Joomla! está hecho con código abierto, lo cual quiere decir que lo puede modificar para hacerlo funcionar del modo que usted desee.</p>
        ]>

```

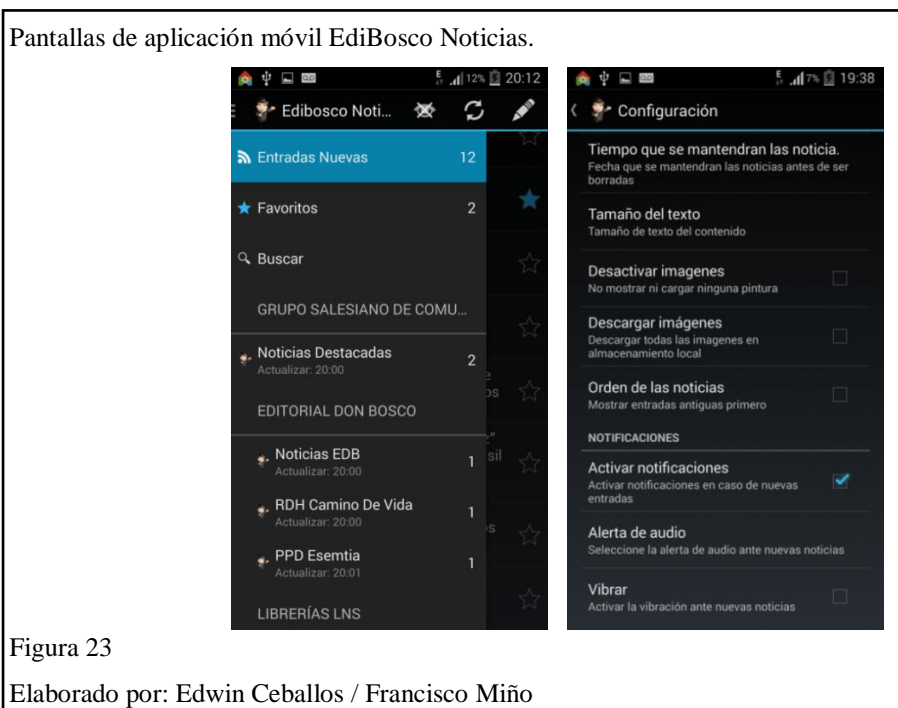
```

<p> &nbsp;</p>
]]>
</desc>
<id>24</id>
</channel>
</item>

```

4.4 Construcción de interfaces de usuario

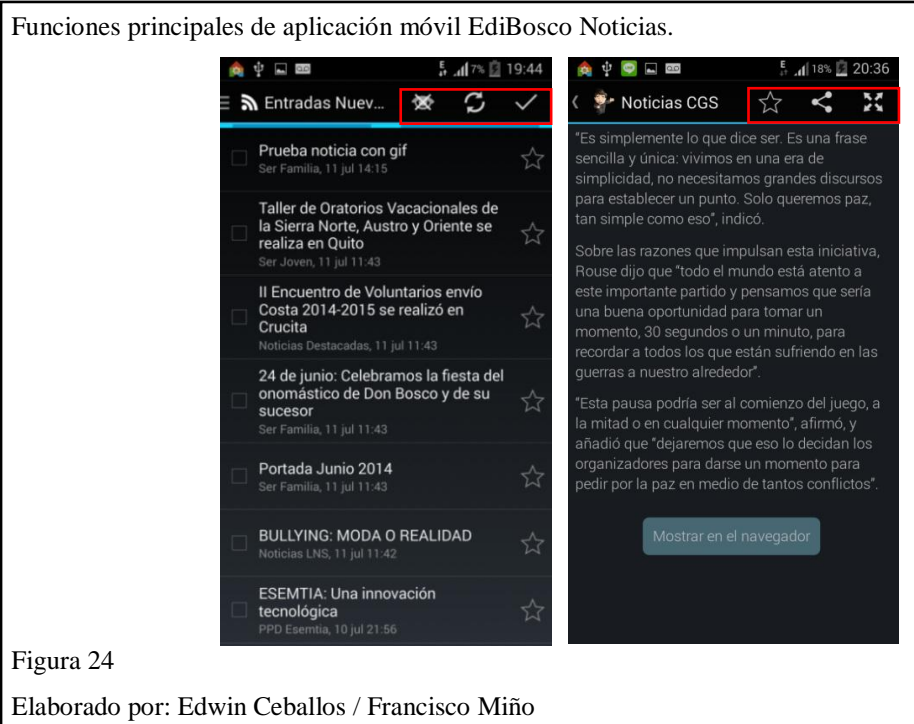
La interface de la aplicación móvil es el medio con que el usuario puede comunicarse con el dispositivo móvil y comprende todos los puntos de contacto entre el usuario y el equipo, se ha diseñado de manera que sea fácil de entender y usar por el usuario, ya que cada una de las opciones tiene su respectiva explicación. Se han utilizado un conjunto de imágenes y objetos gráficos para representar la información y las acciones disponibles para el usuario, como se indica en la figura 23.



Su principal uso consiste en proporcionar un entorno visual sencillo y ordenado de toda la información para permitir la comunicación con el dispositivo móvil.

Para el uso de esta aplicación móvil las acciones se realizan mediante la manipulación directa de una pantalla táctil, para facilitar la interacción entre el usuario y el dispositivo móvil.

La interface cuenta con funciones muy útiles para los usuarios que navegan por la aplicación móvil, algunas de estas son visualizar, compartir, ordenar y refrescar las noticias, como se indica en los recuadros rojos de la figura 24.



CAPÍTULO 5

PRUEBAS

En este capítulo se detalla el plan de pruebas a seguir para la evaluación final del aplicativo móvil, se identifican los tipos de pruebas realizados con su respectiva explicación, se detalla los resultados de dichas pruebas y se indican los pasos realizados para la publicación e instalación de la aplicación móvil.

5.1 Plan de pruebas

Es un documento donde se explican los propósitos y enfoques del proceso de prueba de un sistema, como son: el plan de trabajo, los procedimientos operacionales, las herramientas requeridas y ámbito de responsabilidades.

Seguidamente se muestra un marco de referencia del contenido de un plan de pruebas de proyectos de software:

- Establecimiento del plan de pruebas y del software a evaluar.
- Elementos a probar: casos de uso a probar.
- Calendario: tiempos en el proceso (Myers & John, 2004).

El presente documento describe los métodos usados para verificar que la solución propuesta satisface la especificación requerida y las necesidades del cliente. Incluye los tipos de prueba, las herramientas de software, el ambiente de pruebas y el cronograma de trabajo.

Tabla 12

Plan de pruebas.

Tipo de prueba	Descripción de prueba	Herramienta utilizada
Pruebas funcionales	<p>Las pruebas funcionales son un proceso que permite encontrar discrepancias entre el software desarrollado y/o configurado y la especificación funcional. La prueba funcional es una actividad de caja negra y permite validar:</p> <ul style="list-style-type: none"> • Los procesos y reglas de negocio establecidas. • Que se cumplan los requerimientos funcionales definidos. <p style="text-align: center;">Técnicas de ejecución</p> <ul style="list-style-type: none"> • Crear los casos de prueba. • Ejecutar los casos de prueba. • Reportar los defectos encontrados según las pruebas. <p>En esta prueba se validaron los casos de uso que fueron aprobados por el cliente, y a partir de ellos se diseñan y ejecutan los casos de pruebas correspondientes.</p>	<p>No se utiliza software, debido a que este tipo de prueba verifica la implementación de los casos de uso en la solución propuesta de forma manual.</p>
Pruebas no funcionales	<p>Las pruebas no funcionales garantizan el correcto desempeño de la aplicación móvil en todo tipo de dispositivos móviles, lo que ayuda a que tenga una mejor aceptación por parte de los usuarios, enfatizando los siguientes atributos: Fiabilidad, Portabilidad, Rendimiento, Mantenibilidad, Usabilidad, Disponibilidad y Seguridad, detallados en el punto 3.1.1.</p> <p style="text-align: center;">Técnica de ejecución</p> <ul style="list-style-type: none"> • Crear los casos de prueba. • Ejecutar los casos de prueba. • Reportar los defectos encontrados según las pruebas. 	<p>No es necesario utilizar software, debido a que este tipo de prueba verifica el cumplimiento de los requerimientos no funcionales de forma manual, pero se puede apoyar con la herramienta Android Virtual Device (AVD Manager).</p> <p style="text-align: center;">Descripción</p> <p>Permite emular en una computadora el comportamiento de la aplicación móvil en dispositivos móviles.</p>

Tipo de prueba	Descripción de prueba	Herramienta utilizada
<p>Pruebas de rendimiento del servicio web</p>	<p>La prueba de rendimiento del servicio web se lleva a cabo para asegurar de que la aplicación bajo prueba reunirá los datos de los usuarios en el entorno de producción, además envía cargas de prueba a nivel mundial con distintas IP.</p> <p>Técnica de ejecución</p> <ul style="list-style-type: none"> Probar la dirección web donde se encuentra el descriptor del servicio web EdiBosco. 	<p>Pingdom Tools.</p> <p>Descripción</p> <p>Permite analizar el desempeño de la dirección web (http://Inscompras.com/apk/rss.php) donde se encuentra el descriptor del servicio web EdiBosco (descrito en el punto 4.2.1) para determinar e identificar los potenciales problemas que pueden estar afectando su velocidad de carga.</p>

Nota: Detalle general

Elaborado por: Edwin Ceballos / Francisco Miño

5.1.1 Ambiente de pruebas

Las pruebas realizadas cumplieron con los requerimientos no funcionales y funcionales antes expuestos, se puede mencionar que se realizó pruebas en los siguientes dispositivos móviles:

- **Tablet Samsung Galaxy Tab 2**
 - **Memoria:** 16 GB.
 - **Pantalla:** TFT PLS Touchscreen capacitivo 10.1 pulgadas.
 - **Procesador:** Cortex-A9 Dual-Core 1 GHz.
 - **Memoria RAM:** 1 GB.
 - **Sistema Operativo:** Android 4.0 (Ice Cream Sandwich).
- **Smartphone Samsung Galaxy S3 mini**
 - **Memoria:** 8 GB.
 - **Pantalla:** Super AMOLED 4.0 pulgadas.
 - **Procesador:** NovaThor U8420 Dual-Core 1 GHz.
 - **Memoria RAM:** 1 GB.
 - **Sistema Operativo:** Android 4.1 (Jelly Bean).

5.1.2 Cronograma de trabajo

El cronograma de trabajo planificado para ejecución del plan de pruebas, se muestra a continuación:

Tabla 13

Cronograma de trabajo planificado para la ejecución de pruebas.

Tipo de pruebas	Duración estimada (horas)
Funcionales	10
No funcionales	10
Rendimiento del servicio web	5

Nota: Detalle general

Elaborado por: Edwin Ceballos / Francisco Miño

5.2 Pruebas funcionales

Culminado el proceso de construcción de la solución propuesta, se realizaron las pruebas de caja negra tanto para tareas de usuario como para tareas del webmaster. Estas pruebas se realizaron para los diferentes casos de uso identificados en el punto 3.3.1 de este trabajo, comprobando que el sistema desempeñe el comportamiento

esperado en cada caso de uso y así cumpla con los requerimientos funcionales especificados en el punto 3.1.2, satisfaciendo las necesidades del usuario.

Los casos de prueba han sido registrados en tablas, cuyos campos se detallan a continuación:

- **Código:** Identificador del caso de prueba. El formato definido es: PRFU-001 (número secuencial) para las pruebas de funcionalidad de usuario y PRFW-001 (número secuencial) para las pruebas de funcionalidad del webmaster.
- **Número de caso de prueba:** Determina el orden del caso de prueba.
- **Usuario:** Actor ejecutor de caso de prueba.
- **Referente al caso de uso:** Nombre del caso de uso que está sometido a prueba.
- **Nombre:** Nombre del caso de prueba.
- **Entrada:** Datos, información o salida de otros procesos, los cuales producirán la prueba requerida.
- **Salida:** Resultado o garantía de éxito en el cumplimiento de la prueba.
- **Descripción:** Texto que describe al caso de prueba.
- **Procedimiento de prueba:** Secuencia de pasos para el desarrollo de la prueba.
- **Resultado:** Descripción del comportamiento ideal del aplicativo móvil durante la ejecución del procedimiento de prueba por parte del usuario y webmaster.

Seguidamente se citan los casos de prueba de funcionalidad de usuario y de webmaster definidos:

Tabla 14

Caso de prueba Mostrar al usuario los grupos de noticias de las obras salesianas de forma ordenada.

PRFU-001	
No. caso de prueba	1
Usuario	Usuario
Referente al caso de uso	Ingresar al aplicativo móvil EdiBosco Noticias.
Nombre	Mostrar al usuario los grupos de noticias de las obras salesianas de forma ordenada.
Entrada	Abrir aplicación móvil a través de la función touch.
Salida	Mostrar un listado ordenado y agrupado de las noticias del Grupo Salesiano de Comunicación.
Descripción	Seleccionar un grupo de noticias.
Procedimiento de prueba	1) Abrir la aplicación móvil. 2) Ubicarse en el menú principal. 3) Buscar una obra salesiana. 4) Seleccionar la misma. 5) Visualizar el resultado de la selección.
Resultado	El usuario visualiza un listado ordenado y agrupado de los grupos de noticias del Grupo Salesiano de Comunicación, los cuales son presentados en la sección Índice dentro del aplicativo móvil EdiBosco Noticias.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 15

Caso de prueba Permitir al usuario conocer en que sección del aplicativo móvil se encuentra.

PRFU-002	
No. caso de prueba	2
Usuario	Usuario
Referente al caso de uso	Conocer ubicación dentro del aplicativo móvil EdiBosco Noticias.
Nombre	Permitir al usuario conocer en que sección del aplicativo móvil se encuentra.
Entrada	Abrir aplicación móvil a través de la función touch y seleccionar cualquier sección.
Salida	Visualizar una barra con información de la ubicación dentro de la aplicación móvil.
Descripción	Mostrar la ubicación al usuario.
Procedimiento de prueba	1) Abrir la aplicación móvil. 2) Realizar una selección de cualquier sección. 3) Visualizar en que sección se encuentra dentro del aplicativo móvil.
Resultado	El usuario visualiza en la barra de ubicación en que sección se encuentra.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 16

Caso de prueba Brindar al usuario un conjunto de funciones que permitan visualizar, compartir, ordenar y refrescar las noticias.

PRFU-003	
No. caso de prueba	3
Usuario	Usuario
Referente al caso de uso	Aplicar un conjunto de funciones a las noticias.
Nombre	Brindar al usuario un conjunto de funciones que permitan visualizar, compartir, ordenar y refrescar las noticias.
Entrada	Seleccionar un grupo de noticias del Grupo Salesiano de Comunicación a través de la función touch.
Salida	Visualizar una barra de íconos con funciones que se aplican a noticias individuales o a un grupo de noticias.
Descripción	Visualizar, compartir, ordenar y refrescar la información de las noticias.
Procedimiento de prueba	1) Seleccionar un grupo de noticias del Grupo Salesiano de Comunicación. 2) Elegir una noticia. 3) Seleccionar una función sobre el contenido de la misma.
Resultado	El usuario selecciona una función que se aplica al contenido de las noticias.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 17

Caso de prueba Mostrar al usuario los contenidos de las noticias de manera ordenada.

PRFU-004	
No. caso de prueba	4
Usuario	Usuario
Referente al caso de uso	Mostrar contenido de la información noticiosa.
Nombre	Mostrar al usuario los contenidos de las noticias de manera ordenada.
Entrada	Abrir aplicación móvil a través de la función touch y seleccionar cualquier sección.
Salida	Visualizar la información noticiosa en la sección Contenido.
Descripción	Visualizar contenido de las noticias.
Procedimiento de prueba	1) Abrir aplicación móvil. 2) Seleccionar un grupo de noticias del Grupo Salesiano de Comunicación. 3) Visualizar el contenido de las noticias.
Resultado	El usuario al seleccionar un grupo de noticias visualiza la información noticiosa en la sección Contenido.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 18

Caso de prueba Brindar al usuario un listado de configuraciones aplicables a la información noticiosa.

PRFU-005	
No. caso de prueba	5
Usuario	Usuario
Referente al caso de uso	Aplicar configuraciones a las noticias.
Nombre	Brindar al usuario un listado de configuraciones aplicables a la información noticiosa.
Entrada	Abrir aplicación móvil a través de la función touch.
Salida	Visualizar cambios realizados al activar o desactivar las configuraciones.
Descripción	Activar o desactivar un conjunto de configuraciones.
Procedimiento de prueba	1) Abrir aplicación móvil. 2) Seleccionar sección Configuración. 3) Visualizar los cambios realizados.
Resultado	El usuario visualiza los cambios aplicados a las noticias.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 19

Caso de prueba Brindar al usuario accesos directos para visualización e ingreso de nuevas noticias.

PRFU-006	
No. caso de prueba	6
Usuario	Usuario
Referente al caso de uso	Brindar accesos directos.
Nombre	Brindar al usuario accesos directos para visualización e ingreso de nuevas noticias.
Entrada	Agregar el widget de EdiBosco Noticias al escritorio de Android.
Salida	Elegir widget de contador o de listado de noticias.
Descripción	Acceder al aplicativo móvil de manera directa y visualizar las nuevas noticias existentes.
Procedimiento de prueba	1) Elegir la opción de agregar widget al escritorio de Android. 2) Seleccionar el widget EdiBosco Noticias. 3) Elegir el widget de contador o listado de noticias.
Resultado	El usuario elige el widget que le resulte conveniente para acceder de una manera más rápida al aplicativo móvil.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 20

Caso de prueba Brindar al usuario alertas o notificaciones de nuevas noticias.

PRFU-007	
No. caso de prueba	7
Usuario	Usuario
Referente al caso de uso	Configurar tiempo de sincronización de las noticias.
Nombre	Brindar al usuario alertas o notificaciones de nuevas noticias.
Entrada	Configurar el tiempo de sincronización de las noticias a través de la función touch.
Salida	Visualizar las notificaciones.
Descripción	Configurar el tiempo de sincronización de la información noticiosa.
Procedimiento de prueba	1) Configurar tiempo de sincronización de las noticias. 2) Visualizar la notificación. 3) Seleccionar la notificación.
Resultado	El usuario visualiza en el panel de notificaciones del dispositivo móvil el número de nuevas noticias existentes.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 21

Caso de prueba Administrar noticias.

PRFW-001	
No. caso de prueba	1
Usuario	Webmaster.
Referente al caso de uso	Administrar noticias.
Nombre	Consulta y visualización de las noticias.
Entrada	Ingreso al sistema.
Salida	Consulta de noticias.
Descripción	Consultar las noticias almacenadas en la base de datos del sistema para su correspondiente administración.
Procedimiento de prueba	1) Ingresar al sitio web EdiBosco. 2) Seleccionar una obra salesiana.
Resultado	El sistema muestra al webmaster el resultado de las noticias.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 22

Caso de prueba Ingresar nueva noticia.

PRFW-002	
No. caso de prueba	2
Usuario	Webmaster.
Referente al caso de uso	Ingresar nueva noticia.
Nombre	Ingreso de una noticia en el sistema.
Entrada	Ingreso al sistema. Detalle de noticia: Título, Fecha, Imagen, Detalle.
Salida	Ingreso de nueva noticia con los ítems especificados en Entrada.
Descripción	Se ingresa una nueva noticia con la información requerida.
Procedimiento de prueba	1) Ingresar al sitio web EdiBosco. 2) Seleccionar una obra salesiana. 3) Dar clic en el botón Nuevo. 4) En los campos respectivos ingresar lo datos indicados en el ítem entrada. 5) Dar clic en el botón Aceptar.
Resultado	El sistema valida la información e ingresa la nueva noticia en la base de datos del sistema.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 23

Caso de prueba Editar noticia.

PRFW-003	
No. caso de prueba	3
Usuario	Webmaster.
Referente al caso de uso	Editar noticia.
Nombre	Edición o visualización del detalle de una noticia.
Entrada	Ingreso al sistema.
Salida	Edición del detalle de noticia.
Descripción	Editar el detalle de una noticia previamente seleccionada y que se encuentra almacenada en la base de datos del sistema.
Procedimiento de prueba	1) Ingresar al sitio web EdiBosco. 2) Seleccionar una obra salesiana. 3) Seleccionar una noticia. 4) Dar clic en el botón Editar.
Resultado	El sistema muestra al webmaster el detalle de la noticia seleccionada.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 24

Caso de prueba Modificar noticia.

PRFW-004	
No. caso de prueba	4
Usuario	Webmaster.
Referente al caso de uso	Modificar noticia.
Nombre	Modificación de una noticia seleccionada.
Entrada	<p>Ingreso al sistema.</p> <p>Detalle de noticia:</p> <ul style="list-style-type: none"> • Título. • Fecha. • Imagen. • Detalle.
Salida	Modificación de datos de noticia en los ítems especificados en Entrada.
Descripción	Se modifica los datos de una noticia ingresada anteriormente con nuevos datos.
Procedimiento de prueba	<ol style="list-style-type: none"> 1) Ingresar al sitio web EdiBosco. 2) Seleccionar una obra salesiana. 3) Seleccionar una noticia. 4) Dar clic en el botón Editar. 5) En los campos respectivos ingresar lo datos indicados en el ítem entrada. 6) Dar clic en el botón Modificar. 7) Dar clic en el botón Aceptar.
Resultado	El sistema valida la información y actualiza los datos de la noticia en la base de datos del sistema.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 25

Caso de prueba Eliminar noticia.

PRFW-005	
No. caso de prueba	5
Usuario	Webmaster.
Referente al caso de uso	Eliminar noticia.
Nombre	Eliminación de una noticia seleccionada.
Entrada	Ingreso al sistema.
Salida	Eliminación de datos de noticia.
Descripción	Se elimina los datos de una noticia ingresada anteriormente.
Procedimiento de prueba	1) Ingresar al sitio web EdiBosco. 2) Seleccionar una obra salesiana. 3) Seleccionar una noticia. 4) Dar clic en el botón Editar. 5) Dar clic en el botón Eliminar. 6) Dar clic en el botón Aceptar para confirmar la eliminación de la noticia.
Resultado	El sistema elimina los datos de la noticia en la base de datos del sistema.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

5.3 Pruebas no funcionales

Los casos de prueba han sido registrados en tablas, cuyos campos se detallan a continuación:

- **Código:** Identificador del caso de prueba. El formato definido es: PRNF-001 (número secuencial).
- **Número de caso de prueba:** Determina el orden del caso de prueba.
- **Nombre:** Nombre del caso de prueba.
- **Descripción:** Texto que describe al caso de prueba.
- **Procedimiento de prueba:** Secuencia de pasos para el desarrollo de la prueba.
- **Resultado:** Descripción del comportamiento de la aplicación móvil durante la ejecución del procedimiento de prueba.

Seguidamente se citan los casos de prueba no funcionales definidos:

Tabla 26

Caso de prueba Rendimiento.

PRNF-001	
No. caso de prueba	1
Nombre	Rendimiento.
Descripción	Definir el tiempo de respuesta en la ejecución del aplicativo móvil.
Procedimiento de prueba	<p>1) Se procedió a simular el ingreso a la aplicación 15 veces (una cada media hora) durante 10 horas, con la herramienta Android Virtual Device (AVD Manager), con un tiempo de carga de información de 1.5 segundos.</p> <p>2) Con la colaboración de 15 empleados de Editorial Don Bosco se procedió a probar la aplicación con un tiempo de carga de 1 segundo, para esto se usó el protocolo de comunicaciones HyperText Transfer Protocol (HTTP) de forma que permite al usuario realizar transacciones mediante el acceso al aplicativo móvil.</p>
Resultado	En conclusión, el tiempo de respuesta al momento de ejecutar la aplicación móvil es de 1 segundo.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 27

Caso de prueba Seguridad.

PRNF-002	
No. caso de prueba	2
Nombre	Seguridad.
Descripción	Definir la integridad y la privacidad de la información almacenada tanto en el aplicativo móvil como en la base de datos.
Procedimiento de prueba	1) Se procedió a crear un servicio web para permitir la interoperabilidad entre el aplicativo web y la aplicación móvil, ya que utiliza un fuerte sistema de seguridad en los firewalls porque se encuentra apoyado en la tecnología HyperText Transfer Protocol (HTTP).
Resultado	En conclusión, existe comunicación entre el aplicativo móvil y la aplicación web y ya que no hay autenticación de usuario, no hay restricción en el contenido de las noticias, por lo que cualquier persona tiene acceso a ellas.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 28

Caso de prueba Fiabilidad.

PRNF-003	
No. caso de prueba	3
Nombre	Fiabilidad.
Descripción	Definir la probabilidad de que el aplicativo móvil funcione adecuadamente durante un período determinado bajo condiciones operativas específicas.
Procedimiento de prueba	1) Se procedió a verificar que la información que muestra la aplicación móvil sea de una fuente confiable. 2) Se comprobó que el texto no presenta ningún error al cargar caracteres especiales, es decir se respeta los acentos y las letras ñ.
Resultado	En conclusión, la aplicación móvil es 90% fiable, significa que la información es fidedigna y no existe ningún error de sintaxis (muestra caracteres especiales sin problemas).

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 29

Caso de prueba Disponibilidad.

PRNF-004	
No. caso de prueba	4
Nombre	Disponibilidad.
Descripción	Definir el porcentaje de tiempo que el aplicativo móvil puede ser usado en trabajo productivo.
Procedimiento de prueba	1) Se procedió a instalar el aplicativo móvil en el dispositivo móvil como una aplicación nativa. 2) Se ingresó a la misma en diferentes horarios (06H00, 12H00, 24H00), sin presentar problemas de carga.
Resultado	En conclusión, la aplicación móvil está disponible el 100% del tiempo y con conexión a Internet, la información se actualiza en tiempo real.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 30

Caso de prueba Mantenibilidad.

PRNF-005	
No. caso de prueba	5
Nombre	Mantenibilidad.
Descripción	Definir la cantidad de esfuerzo requerido para conservar el funcionamiento normal del aplicativo móvil a los cambios presentados durante las versiones o para restituirlo una vez que se ha presentado un evento de falla.
Procedimiento de prueba	1) Se procedió a documentar cada uno de los componentes de software.
Resultado	En conclusión, cada componente de software está debidamente documentado como respaldo para dar mantenimiento a la aplicación.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 31

Caso de prueba Portabilidad.

PRNF-006	
No. caso de prueba	6
Nombre	Portabilidad.
Descripción	Definir la característica que posee el aplicativo móvil para ejecutarse en diferentes plataformas, el código fuente del software es capaz de reutilizarse en vez de crearse un nuevo código cuando el software pasa de una plataforma a otra.
Procedimiento de prueba	1) Se procedió a subir la aplicación móvil a la tienda Google Play. 2) Se procedió a comprobar la compatibilidad con diferentes plataformas del sistema operativo Android.
Resultado	En conclusión, la aplicación móvil se puede descargar desde cualquier parte del mundo y ofrece compatibilidad desde la versión 2.2 de Android.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 32

Caso de prueba Usabilidad.

PRNF-007	
No. caso de prueba	7
Nombre	Usabilidad.
Descripción	Definir la facilidad que los usuarios tienen para interactuar con el aplicativo móvil.
Procedimiento de prueba	1) Se procedió a verificar que los elementos pulsables y el tamaño de fuente sean legibles.
Resultado	En conclusión, la aplicación móvil sigue la figura corporativa de Editorial Don Bosco y es fácil interactuar con la misma.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

5.4 Pruebas de rendimiento del servicio web

EL caso de prueba ha sido registrado en una tabla, cuyos campos se detallan a continuación:

- **Código:** Identificador del caso de prueba. El formato definido es: PRSW-001 (número secuencial).
- **Número de caso de prueba:** Determina el orden del caso de prueba.
- **Nombre:** Nombre del caso de prueba.
- **Descripción:** Texto que describe al caso de prueba.
- **Procedimiento de prueba:** Secuencia de pasos para el desarrollo de la prueba.
- **Resultado:** Descripción del comportamiento del servicio web durante la ejecución del procedimiento de prueba.

Seguidamente se cita el caso de prueba de rendimiento del servicio web definido:

Tabla 33

Caso de prueba Rendimiento del servicio web.

PRSW-001	
No. caso de prueba	1
Nombre	Rendimiento del servicio web.
Descripción	Analizar el desempeño de la dirección web donde se encuentra el descriptor del servicio web.
Procedimiento de prueba	1) Se procedió simular la velocidad de carga del servicio web con la herramienta Pingdom Tools 10.080 veces (una cada minuto) desde diferentes partes del mundo durante 7 días, con un tiempo de carga de 474 milisegundos, esto sirve para determinar e identificar los problemas que pueden estar afectando la velocidad de carga del servicio web.
Resultado	En conclusión, la velocidad de carga del servicio web es satisfactoria.

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

5.5 Resultado de pruebas

A continuación se detallan los resultados de las pruebas funcionales (caja negra), no funcionales y de rendimiento del servicio web, definidos en el plan de pruebas.

5.5.1 Resultado de pruebas funcionales

Una vez definido el plan de pruebas; con todos los casos de prueba y sus respectivos procedimientos de prueba, se procedió a ejecutarlos con la participación de un usuario.

Los criterios de estado especificados para los resultados de esta prueba son: *cerrado* y *pendiente*. Donde, *cerrado* (satisfactorio) indica que el resultado de ejecutar los pasos de una actividad ha sido correcto, y *pendiente* requiere de un reconocimiento más a fondo (revisión de código).

Los resultados obtenidos han sido registrados en una tabla, cuyos campos se detallan a continuación:

- **Perfil:** Tipo de usuario.
- **Caso de prueba:** Número y nombre del caso de prueba.
- **Resultado:** Descripción del comportamiento del sistema con la participación del usuario y webmaster.
- **Estado:** Criterio de estado.

- **Observación:** Resultado del comportamiento del sistema.

Tabla 34

Resultados obtenidos enfocados en la funcionalidad del sistema con la participación del usuario.

Perfil	Usuario
Caso de prueba	1: Mostrar al usuario los grupos de noticias de las obras salesianas de forma ordenada.
Resultado	Usuario ejecuta proceso sin inconvenientes.
Estado	Cerrado (satisfactorio).
Observación	La aplicación móvil cumple con el requerimiento.
Caso de prueba	2: Conocer ubicación dentro del aplicativo móvil EdiBosco Noticias.
Resultado	Usuario ejecuta proceso sin inconvenientes.
Estado	Cerrado (satisfactorio).
Observación	La aplicación móvil cumple con el requerimiento.
Caso de prueba	3: Brindar al usuario un conjunto de funciones que permitan visualizar, compartir, ordenar y refrescar las noticias.
Resultado	Usuario ejecuta proceso sin inconvenientes.
Estado	Cerrado (satisfactorio).
Observación	La aplicación móvil cumple con el requerimiento.
Caso de prueba	4: Mostrar al usuario los contenidos de las noticias de manera ordenada.
Resultado	Usuario ejecuta proceso sin inconvenientes.
Estado	Cerrado (satisfactorio).
Observación	La aplicación móvil cumple con el requerimiento.
Caso de prueba	5: Brindar al usuario un listado de configuraciones aplicables a la información noticiosa.
Resultado	Usuario ejecuta proceso sin inconvenientes.
Estado	Cerrado (satisfactorio).
Observación	La aplicación móvil cumple con el requerimiento.
Caso de prueba	6: Brindar al usuario accesos directos para visualización e ingreso de nuevas noticias.
Resultado	Usuario ejecuta proceso sin inconvenientes.
Estado	Cerrado (satisfactorio).
Observación	La aplicación móvil cumple con el requerimiento.
Caso de prueba	7: Brindar al usuario alertas o notificaciones de nuevas noticias.
Resultado	Usuario ejecuta proceso sin inconvenientes.
Estado	Cerrado (satisfactorio).
Observación	La aplicación móvil cumple con el requerimiento.
Conclusión final de evaluación	CERRADO (satisfactorio).

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 35

Resultados obtenidos enfocados en la funcionalidad del sistema con la participación del webmaster.

Perfil	Webmaster
Caso de prueba	1: Ingresar nueva noticia.
Resultado	Webmaster ejecuta proceso sin inconvenientes.
Estado	Cerrado (satisfactorio).
Observación	La aplicación web cumple con el requerimiento.
Caso de prueba	2: Editar noticia.
Resultado	Webmaster ejecuta proceso sin inconvenientes.
Estado	Cerrado (satisfactorio).
Observación	La aplicación web cumple con el requerimiento.
Caso de prueba	3: Modificar noticia.
Resultado	Webmaster ejecuta proceso sin inconvenientes.
Estado	Cerrado (satisfactorio).
Observación	La aplicación web cumple con el requerimiento.
Caso de prueba	4: Eliminar noticia.
Resultado	Webmaster ejecuta proceso sin inconvenientes.
Estado	Cerrado (satisfactorio).
Observación	La aplicación web cumple con el requerimiento.
Conclusión final de evaluación	CERRADO (satisfactorio).

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

5.5.2 Resultado de pruebas no funcionales

Los resultados obtenidos han sido registrados en una tabla, cuyos campos se detallan a continuación:

- **Perfil:** Tipo de usuario.
- **Caso de prueba:** Número y nombre del caso de prueba.
- **Resultado:** Descripción del comportamiento del sistema.
- **Estado:** Criterio de estado.
- **Observación:** Resultado del comportamiento del sistema.

Tabla 36

Resultados obtenidos enfocados en los requerimientos no funcionales.

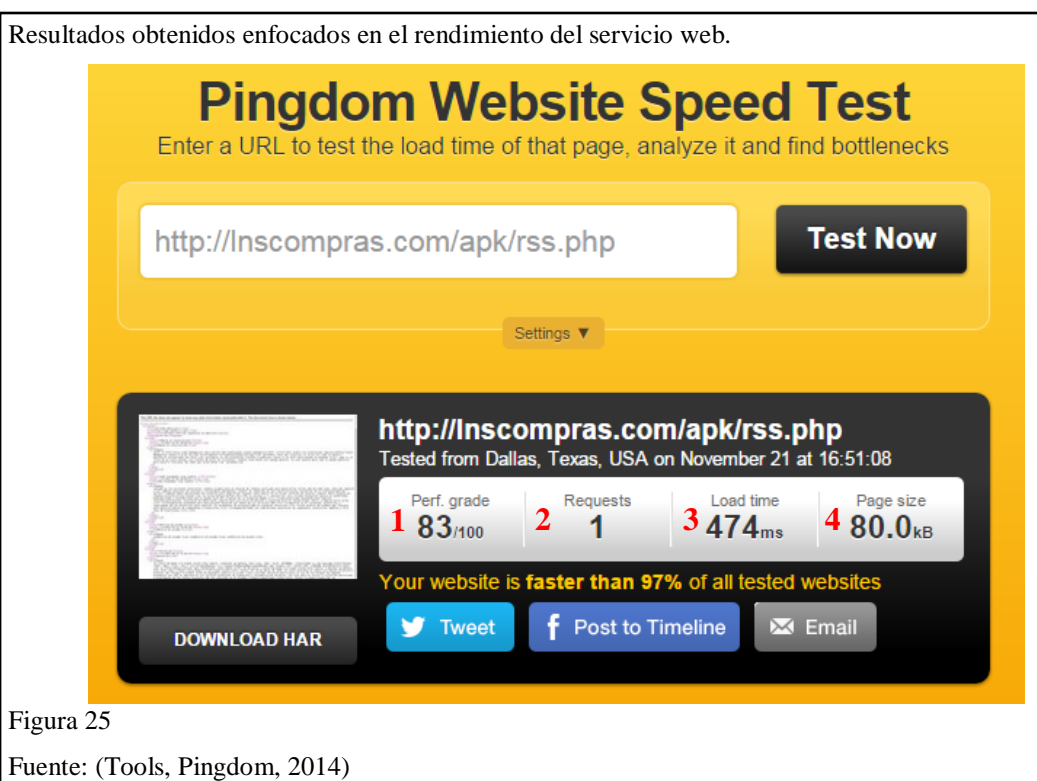
Perfil	Aplicación móvil
Caso de prueba	1: Rendimiento.
Resultado	El tiempo de respuesta al momento de ejecutar la aplicación móvil es de 1 segundo.
Estado	Cerrado (satisfactorio).
Observación	La aplicación móvil cumple con el requerimiento.
Caso de prueba	2: Seguridad.
Resultado	Existe comunicación entre el aplicativo móvil y la aplicación web.
Estado	Cerrado (satisfactorio).
Observación	Se cumple con el requerimiento.
Caso de prueba	3: Fiabilidad.
Resultado	La aplicación móvil es 90% fiable, significa que la información es fidedigna y no existe ningún error de sintaxis (muestra caracteres especiales sin problemas).
Estado	Cerrado (satisfactorio).
Observación	La aplicación móvil cumple con el requerimiento.
Caso de prueba	4: Disponibilidad.
Resultado	La aplicación móvil está disponible el 100% del tiempo y con conexión a Internet, la información se actualiza en tiempo real.
Estado	Cerrado (satisfactorio).
Observación	La aplicación móvil cumple con el requerimiento.
Caso de prueba	5: Mantenibilidad.
Resultado	Cada componente de software está debidamente documentado como respaldo para dar mantenimiento a la aplicación.
Estado	Cerrado (satisfactorio).
Observación	Se cumple con el requerimiento.
Caso de prueba	6: Portabilidad.
Resultado	La aplicación móvil se puede descargar desde cualquier parte del mundo y ofrece compatibilidad desde la versión 2.2 de Android.
Estado	Cerrado (satisfactorio).
Observación	La aplicación móvil cumple con el requerimiento.
Caso de prueba	7: Usabilidad.
Resultado	La aplicación móvil sigue la figura corporativa de Editorial Don Bosco y es fácil interactuar con la misma.
Estado	Cerrado (satisfactorio).
Observación	La aplicación móvil cumple con el requerimiento.
Conclusión final de evaluación	CERRADO (satisfactorio).

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

5.5.3 Resultado de pruebas de rendimiento del servicio web

La herramienta Pingdom Tools utilizada para analizar el desempeño de la dirección web (<http://Inscompras.com/apk/rss.php>) donde se encuentra el descriptor del servicio web EdiBosco, arroja 4 resultados, los cuales se indican en la figura 25.



1. **Grado de rendimiento:** Muestra un análisis sobre el desempeño de la velocidad de carga del servicio web analizado. Como muestra la figura 25, el grado de rendimiento del servicio web EdiBosco es de 83/100, lo cual significa que el resultado es aceptable (Tools, Pingdom, 2014).
2. **Llamadas realizadas:** Muestra el número de llamadas registradas en la carga del servicio web analizado. Como muestra la figura 25, al momento de realizar la prueba, primero se registró una llamada al servicio web EdiBosco, después se registraron 10.080 peticiones (una cada minuto) de diferentes partes del mundo al mismo servicio web durante 7 días, con un promedio de eficiencia del 99.55% (debido a que todos los tiempos de carga están por debajo de los 2 segundos), ya que el servicio solo dejó de funcionar 45 milisegundos, como lo indica la figura 26, lo cual significa que el resultado es satisfactorio, ya que lo recomendable es que las peticiones sean mayores a 50 (Tools, Pingdom, 2014).

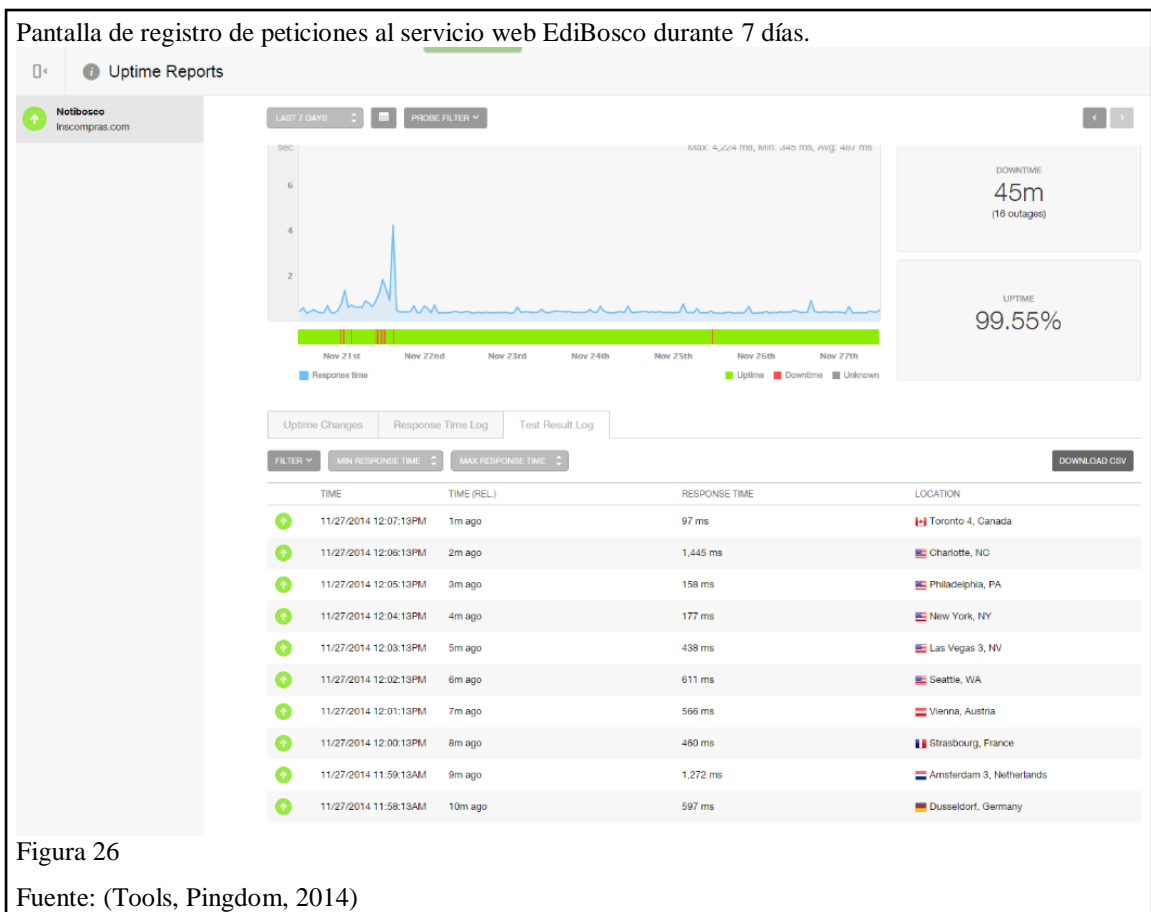


Figura 26

Fuente: (Tools, Pingdom, 2014)

- Tiempo de carga:** Muestra el tiempo total de carga del servicio web analizado. Como muestra la figura 25, el tiempo de carga del servicio web EdiBosco es de 474 milisegundos, lo cual significa que el resultado es satisfactorio, ya que lo recomendable es que el tiempo esté por debajo de los 2 segundos (Tools, Pingdom, 2014).
- Tamaño:** Muestra el peso en KB/MB del servicio web analizado. Como muestra la figura 25, el tamaño del servicio web EdiBosco es de 80.0 KB, lo cual significa que el resultado es satisfactorio, ya que mientras menor sea el espacio que ocupa, es mayor su eficiencia (Tools, Pingdom, 2014).

Además muestra como es de rápido el servicio web EdiBosco respecto al resto de servicios web analizados, en este caso es más rápido que el 97% de servicios web analizados.

A continuación se presenta el resumen de la evaluación de resultados y la evaluación final del sistema:

Tabla 37

Resumen de evaluación de resultados.

Tipo de usuario	Conclusión de evaluación
Webmaster	CERRADO (satisfactorio).
Usuario	CERRADO (satisfactorio).
Aplicación móvil	CERRADO (satisfactorio).
Servicio web	CERRADO (satisfactorio).
Conclusión final del sistema	<p>CERRADO:</p> <p>Existe una concordancia entre los requerimientos funcionales y no funcionales especificados en los puntos 3.1.2 y 3.1.1 respectivamente y tras haber efectuado las pruebas pertinentes donde los usuarios no encontraron mayores inconvenientes en el uso del aplicativo móvil; se ha demostrado que los requerimientos funcionales y no funcionales han sido completamente satisfechos por la aplicación, de modo que el atributo de calidad ha sido cumplido por el aplicativo móvil EdiBosco Noticias.</p>

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

5.6 Publicación del aplicativo en la Web

Luego del desarrollo de la aplicación móvil EdiBosco Noticias, el siguiente paso es la publicación en la tienda Google Play, significa que la aplicación estará disponible a los usuarios a través de Internet y comenzará a funcionar bajo la responsabilidad del usuario según la necesidad del mismo.

Todo este proceso se realizará ingresando información de la Editorial Don Bosco, con el objetivo de que este registro sirva para que futuros proyectos desarrollados dentro de la editorial se publiquen (Ver Anexo 8).

5.7 Descarga e instalación del aplicativo por parte del usuario

Primeramente se debe descargar la aplicación desde la tienda Google Play, para ello hay que dirigirse al menú de aplicaciones del dispositivo móvil, luego se ingresa al gestor de descargas, en este caso se utiliza el Play Store.

En el gestor de descargas se introduce el nombre de la aplicación móvil, se ingresa el texto “EdiBosco”. Se desplegarán todas las aplicaciones que coincidan con la

búsqueda, en este caso se escoge la primera coincidencia. Se selecciona la aplicación EdiBosco Noticias.

Posteriormente se despliega una ventana con la descripción de la aplicación EdiBosco Noticias, luego se presiona la opción instalar.

Se procede a aceptar y descargar la aplicación, cabe recalcar que no se necesita ningún tipo de permisos adicionales.

Una vez completada la descarga de la aplicación, la misma se instalará automáticamente en el dispositivo móvil, finalmente se presiona la opción abrir para ingresar a la aplicación móvil EdiBosco Noticias (Ver Anexo 9).

Conclusiones

- El desarrollo del sistema para la distribución de noticias basado en dispositivos móviles, en sus etapas de análisis, diseño, construcción e implementación, denominado EdiBosco Noticias, ha finalizado satisfactoriamente, debido a que cumple con los requerimientos funcionales, no funcionales y alcance establecidos en un inicio.
- La presentación de noticias a través del dispositivo móvil, alrededor de diferentes tópicos y con contenido multimedia, hace que el usuario móvil tenga mayor interés en usar la aplicación, mismo que significa un aspecto muy importante a la hora de poner en producción un sistema de software.
- La navegabilidad del sistema EdiBosco Noticias facilita al usuario la visualización de noticias acerca de las novedades y de los servicios que ofrece el Grupo Salesiano de Comunicación, ya que la interfaz es amigable, fácil de comprender, por lo que la aplicación puede ser usada sin ninguna preparación previa por parte del usuario.
- Mobile-D, metodología utilizada en la ejecución de este trabajo, presenta algunas ventajas significativas en el desarrollo de una aplicación móvil, ya que se apoya en otras soluciones metodológicas bien conocidas y consolidadas como son eXtreme Programming (XP), Crystal Methodologies y Rational Unified Process (RUP); es una metodología ágil con ciclos de desarrollo cortos y para equipos pequeños y su desarrollo, basado en pruebas, es una de las mejores formas de asegurar la calidad; los casos de uso son determinados de acuerdo a los requerimientos del sistema, permite la reutilización de componentes y un proceso ordenado.
- La división de la aplicación en capas permite un fácil mantenimiento de la misma, razón por la cual en un futuro se puede añadir nuevas funcionalidades en forma modular, así como de cambiar las reglas del negocio de manera transparente para el usuario que utiliza la aplicación móvil.
- El tamaño de la pantalla es reducido en varios ejemplares y modelos de dispositivos móviles. Esta característica obliga a crear aplicaciones sencillas y al mismo tiempo organizadas y fáciles de comprender para el usuario.
- Normalmente el costo de una gran cantidad de modelos de dispositivos móviles es menor al costo de un computador personal lo cual prevé que en un

futuro próximo la penetración que en el mercado se va a tener respecto al uso de este tipo de dispositivos será muy significativa, ya que los mismos servicios podrán accederse desde cualquier lugar gracias al uso de este tipo de tecnologías con una relación costo-beneficio mucho mayor respecto del uso de tecnologías Internet convencionales de escritorio.

- Con la combinación de la notación visual UML, con la metodología Mobile-D, se pudo saber qué aspectos esenciales hay que modelar (desde un bosquejo a un plano detallado).
- Al inicio de este proyecto no se contempló la creación de un servicio web, pero se necesitó crear uno para la interoperabilidad entre el aplicativo web y la aplicación móvil, debido a que es un componente de software de gran utilidad porque permite la comunicación entre el aplicativo móvil y la aplicación web.

Recomendaciones

- Es importante antes de iniciar el proceso de desarrollo de un sistema de software y en particular de este tipo de aplicación, tener una estrategia de mercado, para saber cuáles y cuántos usuario que tienen un dispositivo móvil están interesados en tener una aplicación de tales características y así presentar al usuario sólo información que verdaderamente le sirva en su accionar diario.
- El uso de la metodología Mobile-D en desarrollos de aplicaciones para dispositivos móviles simplifica la concepción de los problemas y brinda una secuencia de pasos fáciles de seguir y se aplica en los procesos de análisis y diseño.
- Emplear software gratuito como los que se ha utilizado para realizar este trabajo de grado es importante, ya que es una solución para bajar costos mínimos en la construcción de una aplicación móvil.
- Es importante que los desarrolladores aprovechen el auge de la tecnología móvil en el país y la aceptación de las variadas aplicaciones de este tipo, para proveer a los usuarios de aplicaciones móviles útiles y comercialmente factibles.

- Adquirir bibliografía actualizada sobre las tecnologías orientadas a desarrollar aplicaciones para dispositivos móviles es importante, ya que es de gran ayuda el tener fuentes bibliográficas que den soporte a futuros proyectos similares.
- Para la generación de los artefactos en las fases de análisis, diseño e implementación es recomendable utilizar el estándar UML porque es diseñado para modelar software orientado a objetos y es independiente de las características de los proyectos.

Lista de referencias

1. Álvarez, P. M. (2009). *Ingeniería de Software*. Lima: Editora ABC Perú.
2. Andrade, A. N. (2010). *Evaluación de la plataforma Android para dispositivos móviles*. Guayaquil: Universidad de Guayaquil.
3. Arco, R. N. (2012). *Aplicativo para dispositivos móviles sobre información turística y hotelera de la ciudad de Guayaquil*. Guayaquil: Universidad Católica Santiago de Guayaquil.
4. Beck, K. (2000). *Extreme Programming Explained*. Londres: EmbraceChange.
5. Cantero, P. C. (2011). *Desarrollo de una aplicación Web para la gestión de entornos virtuales*. Madrid: Universidad Complutense de Madrid.
6. Chacón, L. T. (2013). *Diseño de una aplicación móvil para consulta académica de la FIIS-UTP*. Lima: Universidad Tecnológica del Perú.
7. Cockburn, A. (2004). *A Human-Powered Methodology for Small Teams*. Londres: Crystal Clear.
8. Cuenca, M. B. (2007). Sistema para la publicación de noticias basado en dispositivos de telefonía móvil. Quito: Escuela Politécnica Nacional.
9. Dattatec. (12 de Marzo de 2014). *Dattamagazine*. Obtenido de <http://www.dattamagazine.com>
10. Daybis, I. V. (2014). *Uso Educativo de Satélites de Telecomunicaciones*. Murcia: Departamento de Didáctica y Organización Escolar.
11. *EdiBosco*. (2013). Obtenido de <http://www.edibosco.com>
12. Fosch, A. I., & Arellano, R. J. (2006). *Guía de estilo de codificación*. Catalunya: Free Software Foundation.
13. Gutierrez, A. P. (2010). *Implementación de una red industrial basado en ASI-BUS*. Riobamba: Escuela Superior Politécnica de Chimborazo.
14. Kruchten, P. (1999). *The Rational Unified Process*. Londres: Addison-Wesley Professional.

15. Lasso, F. D. (2013). Sistema de información para el campeonato ecuatoriano de fútbol primera A para plataforma Iphone. Quito: Escuela Politécnica Nacional.
16. Lavoe, I. (2009). *Repositorio de Información que deseo mantener siempre accesible y no quiero olvidar*. Obtenido de <http://ivanlavoe.blogspot.com/2009/09/web-services.html>
17. Marroquín, M. L. (2013). *Estudio y Desarrollo de Aplicaciones Móviles Android*. Ibarra: Universidad Técnica del Norte.
18. Myers, & John, W. (2004). *The art of Software Testing*. New York: Prentice Hall PTR.
19. Olsen, H. (2007). *The interaction designer's nail gun*. Obtenido de <http://www.nosolousabilidad.com/articulos/diagramacion.htm>
20. Peña, K. R. (01 de Agosto de 2012). *PC World*. Obtenido de <http://www.pcworld.com.mx/Articulos/24531.htm>
21. Peralta, D. F. (2012). Diseño e Implementación de una aplicación para dispositivos Android en el marco del proyecto pequeñas y pequeños científicos de la Universidad Politécnica Salesiana. Cuenca.
22. Pérez Reyes, A. (2014). *Análisis de rendimiento de aplicaciones Android*. La Laguna: Universidad de La Laguna.
23. Petrazzini, G. O. (2012). *Sistemas operativos en dispositivos móviles*. Buenos Aires: Universidad Nacional del Nordeste.
24. Ruiz, E. A., & Estévez, L. E. (2012). *Análisis, diseño, implementación e implantación del sistema WEB para manejo de hospitalización, consulta externa y centro quirúrgico en la nueva Clínica Bolívar*. Quito: Universidad Politécnica Salesiana.
25. Tierra, G. P. (2010). *Estudio comparativo entre herramientas AMP (Apache, MySQL, PHP)*. Riobamba: Escuela Superior Politécnica de Chimborazo.
26. Tools, Pingdom. (20 de Noviembre de 2014). *Pingdom Tools*. Obtenido de <http://tools.pingdom.com/fpt/>

27. Tudela, J. A. (2009). *Desarrollo de aplicaciones para dispositivos móviles sobre la plataforma Android de Google*. Madrid: Universidad Carlos III de Madrid.
28. Wayerless. (2012). *Sistemas operativos móviles mas usados en Ecuador*. La Paz: Iberoamérica.
29. Yorio, D. (2011). *Identificación y clasificación de patrones en el diseño de aplicaciones móviles*. La Plata: Universidad Nacional de la Plata.

Anexos

Anexo 1. Encuesta realizada a 20 personas para conocer el detalle de los servicios que ofrece el Grupo Salesiano de Comunicación.

1. ¿Cuáles son las actividades empresariales en las que se utiliza aplicaciones informáticas?

- Consulta de inventarios.
- Compra en línea. 8 Personas: Compra en línea → 40%
- Manejo de información. Resultados: 5 Personas: Noticias → 25%
- Facturación. 4 Personas: Facturación → 20%
- Noticias. 3 Personas: Manejo de información → 15%
- Otro (Por favor especifique).

2. ¿Qué tipo de licenciamiento tienen los sistemas utilizados en la empresa?

- Software bajo licencia. Software bajo licencia → 75%
- Software gratis. Resultados: Software gratis → 25%

3. ¿Quién le provee los programas para equipos de escritorio?

- Los desarrolladores dentro de la empresa.
- Descargas por Internet. Los desarrolladores dentro de la empresa → 80%
- Proveedor local. Resultados: Descargas por Internet → 20%
- Otro (Por favor especifique).

4. ¿La empresa utiliza aplicaciones móviles?

- Si. 12 Personas: Si → 60%
- No. Resultados: 8 Personas: No → 40%

5. ¿La empresa requiere de aplicaciones móviles para el desarrollo de sus actividades?

- Si. 16 Personas: Si → 80%
- No. Resultados: 4 Personas: No → 20%

6. ¿Qué tipo de aplicaciones sobre dispositivos móviles requiere en las actividades de la empresa?

- Consulta de inventarios.
- Compra en línea. 8 Personas: Noticias → 40%
- Manejo de información. Resultados: 5 Personas: Compra en línea → 25%
- Facturación. 4 Personas: Facturación → 20%
- Noticias. 3 Personas: Manejo de información → 15%
- Otro (Por favor especifique).

7. Describa la aplicación para dispositivos móviles que considere requiere la empresa.

El Grupo Salesiano de Comunicación requiere una aplicación móvil de noticias ya que es importante estar actualizado/a de las últimas novedades que se suscitan en dicha empresa.

8. ¿Cuál es el motivo para que la empresa no haya utilizado o requerido de una aplicación para dispositivos móviles para el desarrollo de las actividades diarias?

- No necesitan de este tipo de aplicaciones.
- No cuentan con dispositivos móviles.
- Desconocen los beneficios de este tipo de aplicaciones.
- Temor a la innovación de productos y procesos.
- Otro (Por favor especifique).

3 Personas: No necesitan de este tipo de aplicaciones → 15%

Resultados: 2 Personas: No cuentan con dispositivos móviles → 10%

10 Personas: Desconocen los beneficios de este tipo de aplicaciones → 50%

5 Personas: Temor a la innovación de productos y procesos → 25%

9. ¿Cómo fue la experiencia de utilización de dispositivos móviles para la visualización de las noticias?

- | | | |
|--------------|-------------|------------------------------|
| • Excelente. | | 14 Personas: Excelente → 70% |
| • Buena. | Resultados: | 4 Personas: Buena → 20% |
| • Regular. | | 1 Persona: Regular → 5% |
| • Mala. | | 1 Persona: Mala → 5% |

10. ¿Aumentó el grado de interés en las noticias utilizando dispositivos móviles?

- | | | |
|-------|-------------|-----------------------|
| • Si. | | 16 Personas: Si → 80% |
| • No. | Resultados: | 4 Personas: No → 20% |

11. ¿La utilización de la aplicación EdiBosco Noticias en dispositivos móviles fue?

- | | | |
|------------|-------------|---------------------------|
| • Difícil. | | 15 Personas: Fácil → 75% |
| • Normal. | Resultados: | 3 Personas: Normal → 15% |
| • Fácil. | | 2 Personas: Difícil → 10% |

Anexo 2. Diccionario de datos de EdiBosco Noticias.

Tabla 38

Tabla Entrada.


Nombre	DataType	Anulable
Id	Int	No
Feedid	Int	No
Title	Char	No
Abstract	Char	No
Movilized	Char	No
Date	Date	No
Fetch_date	Date	No
Isread	Int	No
Link	Char	No
Favorite	Int	No
Enclosure	Char	No
Guid	Char	No
Author	Char	No

Nota: Resumen general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 39

Detalles relación FK_ENTRADA_REFERENCE_FILTRO.

FK_ENTRADA_REFERENCE_FILTRO	
To	 FILTRO
Autor	Francisco Miño
Create Date Time	10/07/2014 11:46:39 AM
Last Modified	10/07/2014 11:47:16 AM
Identifying	False
To Multiplicity	0..*
From Multiplicity	1
Sync to Association	Yes
Datos Modelo	Physical

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 40

Tabla Noticia.


Nombre	DataType	Anulable
Id	Int	No
Name	Varchar(10)	No
Isgroup	Int	No
Groupid	Int	No
Lastupdate	Date	No
Reallastupdate	Date	No
Retrievefulltext	Int	No
Icon	Image	No
Error	Char	No
Priority	Int	No
Fetchmode	Int	No

Nota: Resumen general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 41

Detalles relación FK_FILTRO_REFERENCE_NOTICIA.

FK_FILTRO_REFERENCE_NOTICIA	
To	 NOTICIA
Autor	Francisco Miño
Create Date Time	10/07/2014 12:10:39 PM
Last Modified	10/07/2014 12:11:16 PM
Identifying	False
To Multiplicity	0..*
From Multiplicity	1
Sync to Association	Yes
Datos Modelo	Physical

Nota: Descripción general

Elaborado por: Edwin Ceballos / Francisco Miño

Tabla 42

Tabla Filtro.

Nombre	DataType	Anulable
Id	Int	No
Feedid	Int	No
Filtertext	Char	No
Isregex	Int	No
Isappliedtotitle	Int	No
Isacceptrule	Int	No

Nota: Resumen general

Elaborado por: Edwin Ceballos / Francisco Miño

Las entidades en este sistema son las siguientes tablas: Entrada, Noticia y Filtro, los atributos son las columnas de dichas tablas y las claves primarias se convierten en índices en cada una de las tablas.

Anexo 3. Instalación y configuración del entorno de desarrollo JDK.

Para instalar el entorno de desarrollo JDK, se debe seguir los siguientes pasos:

1. Descargar el JDK del sitio web de Oracle <http://www.oracle.com/technetwork/java/javase/downloads/index.html>, en esta página se encuentran las diversas versiones.
2. Se procede a aceptar los términos de licencia y escoger el instalador del sistema operativo en el que se está desarrollando, en este caso se escoge Windows, como lo indica la figura 27.

Pantalla con los términos de licencia del entorno de desarrollo JDK.



Figura 27

Elaborado por: Edwin Ceballos / Francisco Miño

3. Con esto se descargará un archivo ejecutable, en el cual se procede a dar doble clic para la instalación y posteriormente se da clic en *Siguiente*, como lo indica la figura 28.

Pantalla de instalación del entorno de desarrollo JDK.

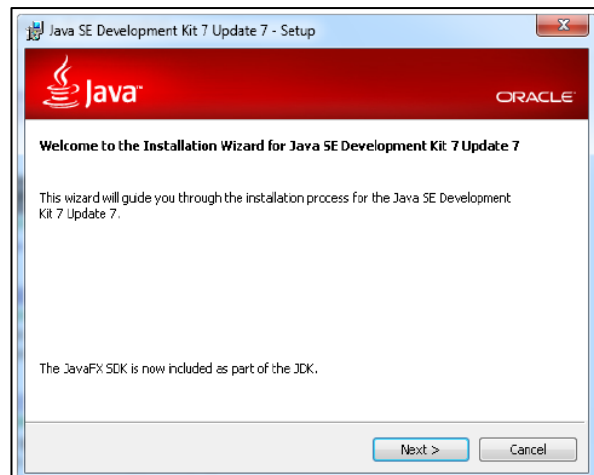


Figura 28

Elaborado por: Edwin Ceballos / Francisco Miño

4. Se escoge las características que se desea instalar y se da clic en *Siguiente*, se procederá a instalar el JDK, como lo indica la figura 29.

Pantalla de características que se desea instalar.

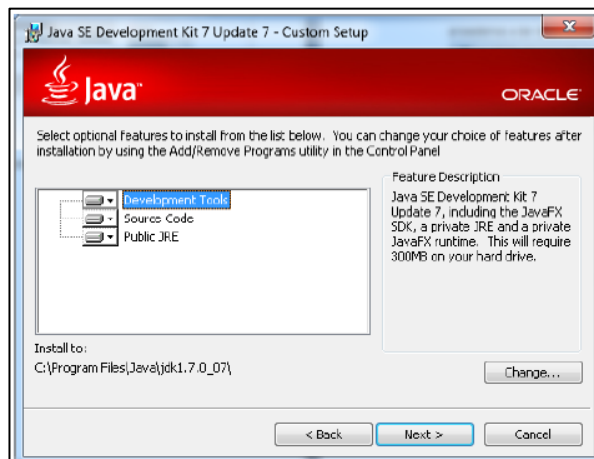


Figura 29

Elaborado por: Edwin Ceballos / Francisco Miño

5. Por último se presenta un mensaje que informa que la instalación se realizó con éxito, se da clic en *Cerrar*, como lo indica la figura 30.

Pantalla de resultado de instalación del entorno de desarrollo JDK.

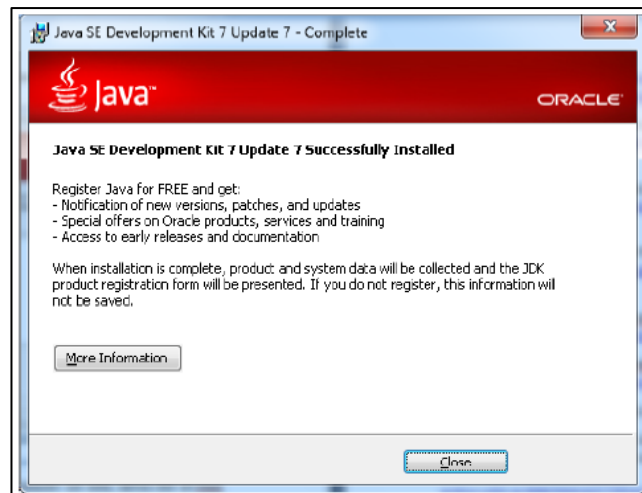


Figura 30

Elaborado por: Edwin Ceballos / Francisco Miño

Anexo 4. Instalación y configuración del SDK de Android.

1. Descargar la versión del SDK de Android del sitio web <http://developer.android.com/sdk/index.html>, como lo indica la figura 31.

Sitio web del instalador del SDK de Android.

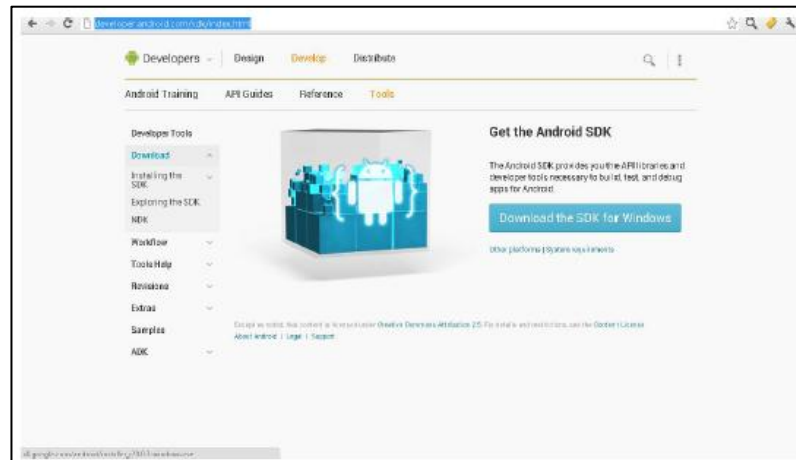


Figura 31

Elaborado por: Edwin Ceballos / Francisco Miño

2. Se procede a instalar el SDK de Android, al dar doble clic en el archivo descargado, se presenta la siguiente pantalla y se da clic en *Siguiente*, como se indica en la figura 32.

Pantalla de instalación del SDK de Android.

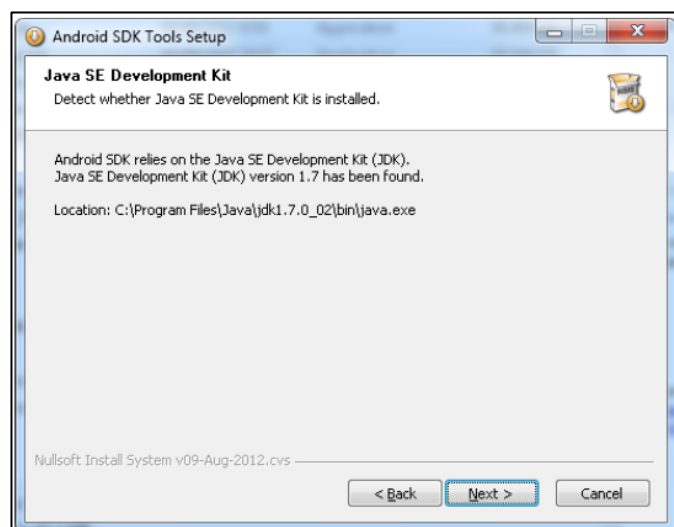


Figura 32

Elaborado por: Edwin Ceballos / Francisco Miño

3. Se escoge el directorio en donde se quiere instalar el SDK y se procede a dar clic en *Siguiente*, como lo indica la figura 33.

Pantalla donde se selecciona el directorio de instalación del SDK de Android.

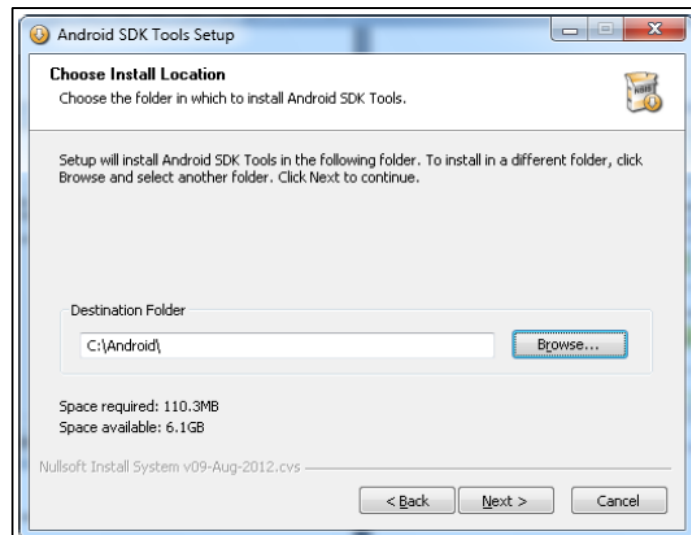


Figura 33

Elaborado por: Edwin Ceballos / Francisco Miño

4. Se procede a instalar todos los paquetes del SDK de Android al dar clic en *Instalar*, como lo indica la figura 34.

Pantalla con los paquetes de SDK de Android.

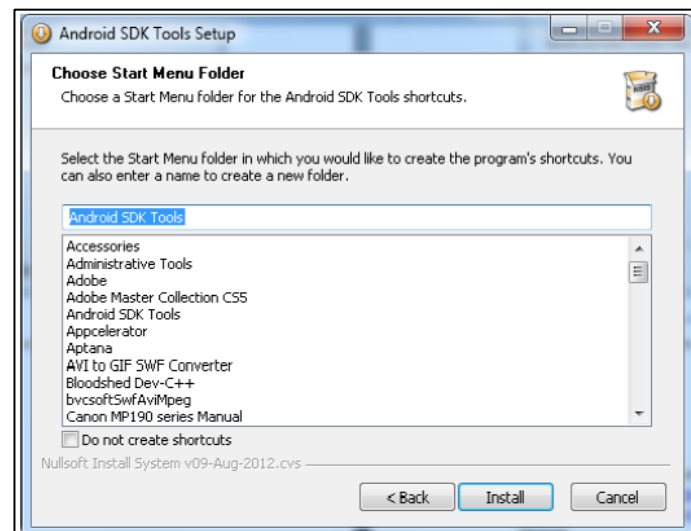


Figura 34

Elaborado por: Edwin Ceballos / Francisco Miño

5. Una vez completada la instalación se procede a dar clic en *Finalizar*.

6. Posteriormente se deben instalar paquetes adicionales del sistema operativo Android, con esto se puede ejecutar la aplicación móvil dentro de un dispositivo virtual, se escoge la última versión de Android como base para realizar las pruebas

en las diferentes versiones, luego se da clic en *Instalar Paquetes*, como lo indica la figura 35.

Pantalla de instalación de paquetes del SDK de Android.

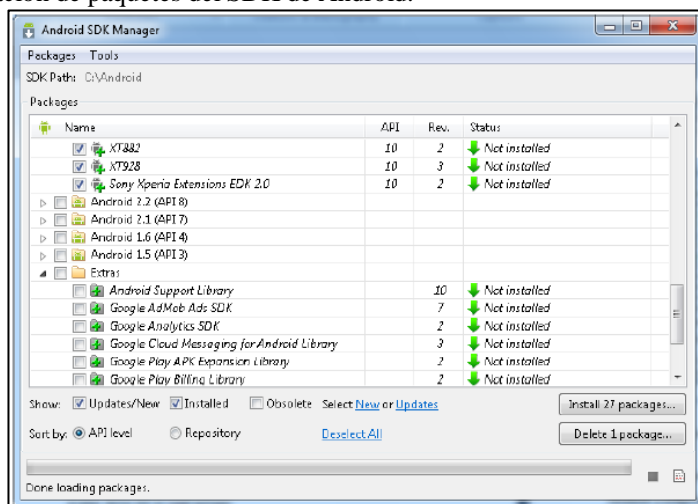


Figura 35

Elaborado por: Edwin Ceballos / Francisco Miño

7. Una vez terminada la instalación de los paquetes se procede a crear una máquina virtual de Android, para esto se selecciona del Menú la opción *Tools* y se escoge *Manager ADVS*, como lo indica la figura 36, aquí se puede crear los dispositivos virtuales para la ejecución de la aplicación en un entorno virtual.

Pantalla con lista de dispositivos virtuales.

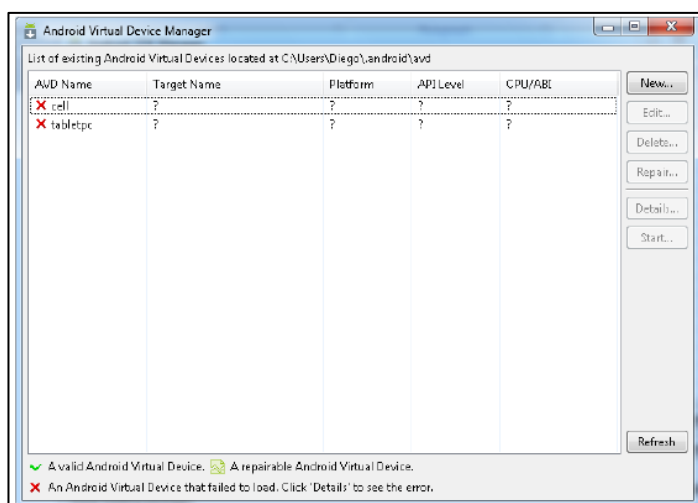


Figura 36

Elaborado por: Edwin Ceballos / Francisco Miño

8. Para crear un nuevo dispositivo virtual se da clic en *Nuevo*, como lo indica la figura 37.

Pantalla de creación de dispositivos virtuales.

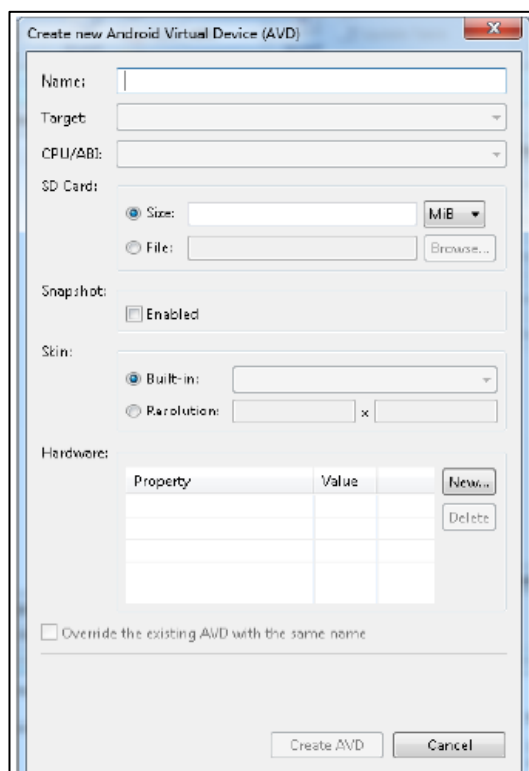


Figura 37

Elaborado por: Edwin Ceballos / Francisco Miño

9. Se ingresa el nombre del dispositivo virtual, la plataforma en la que se va a ejecutar, se asigna el tamaño de memoria, el tipo de pantalla y el hardware con el que cuenta el dispositivo, una vez terminado se da clic en *Create ADV*, como lo indica la figura 38.

Pantalla de ingreso de características del dispositivo virtual.

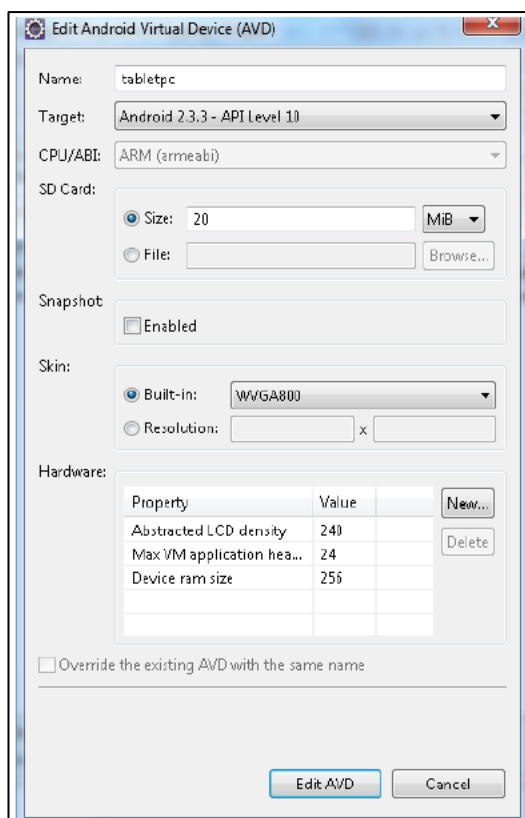


Figura 38

Elaborado por: Edwin Ceballos / Francisco Miño

10. Una vez terminada la creación del dispositivo virtual, se puede iniciarlo para comprobar su correcto funcionamiento, para esto se da clic en *Comenzar*, como lo indica la figura 39.

Pantalla de comprobación de funcionamiento de dispositivo virtual.

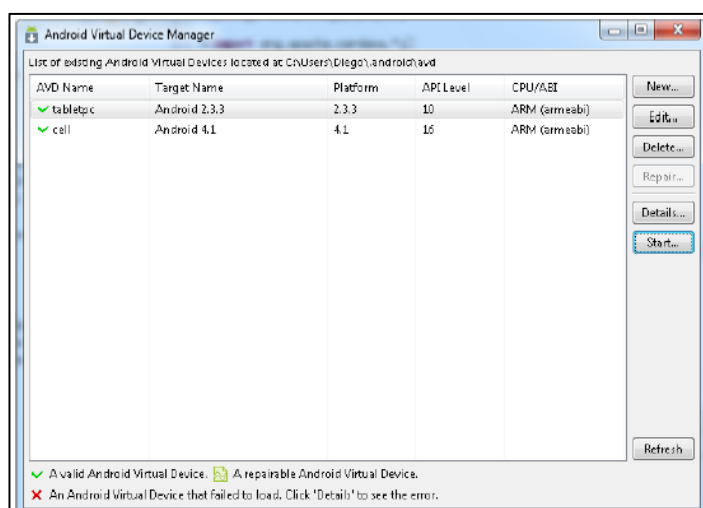


Figura 39

Elaborado por: Edwin Ceballos / Francisco Miño

11. En la figura 40 se muestra la ejecución del dispositivo virtual.

Pantalla de ejecución del dispositivo virtual.



Figura 40

Elaborado por: Edwin Ceballos / Francisco Miño

Anexo 5. Instalación y configuración de Eclipse Juno.

Para instalar el entorno de desarrollo Eclipse, se debe seguir los siguientes pasos:

1. Se utilizó Eclipse Juno 4.2 como entorno de desarrollo, el cual se descargó del sitio <http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/juno/R/eclipse-jee-juno-win32.zip>, dentro de este sitio se escoge el servidor de descarga, como lo indica la figura 41.

Sitio web de descarga de Eclipse Juno 4.2.

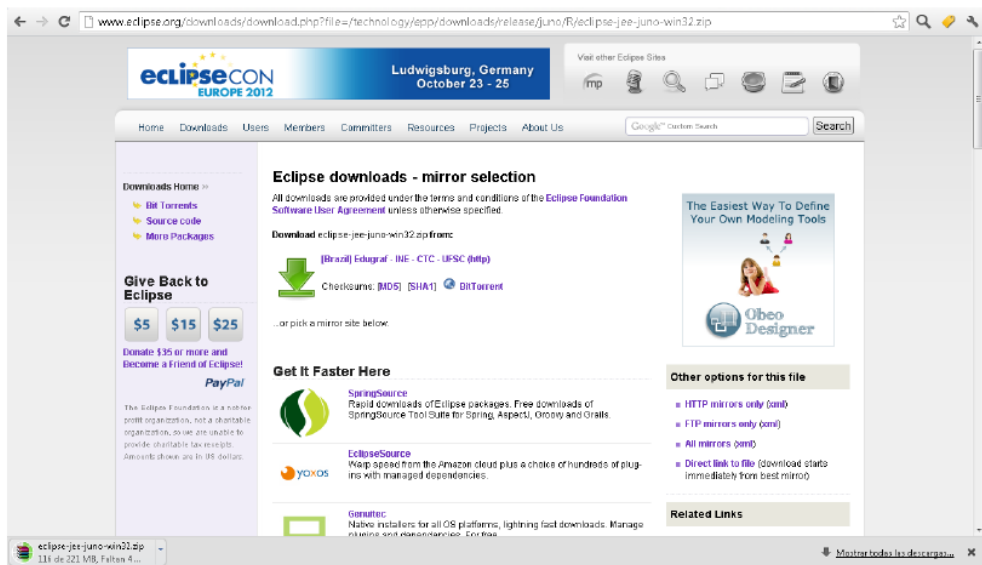


Figura 41

Elaborado por: Edwin Ceballos / Francisco Miño

2. Una vez finalizada la descarga no se puede realizar ningún proceso de instalación, simplemente se debe descomprimir el fichero en un nuevo directorio y se da clic en el ejecutable de Eclipse, la primera vez que se ejecuta la aplicación solicita escoger el espacio de trabajo (workspace), carpeta donde se almacenará la aplicación desarrollada.

Anexo 6. Instalación del plugin de Android para Eclipse.

Para instalar el plugin específico de Android para la plataforma de desarrollo Eclipse es necesario valerse de la herramienta Android Development Tools (ADT) que facilita enormemente el desarrollo de proyectos basados en Android.

1. Seleccionar en el Menú la opción *Help* y luego escoger la opción *Install New Software*, como lo indica la figura 42.

Pantalla de instalación de plugin de Android.

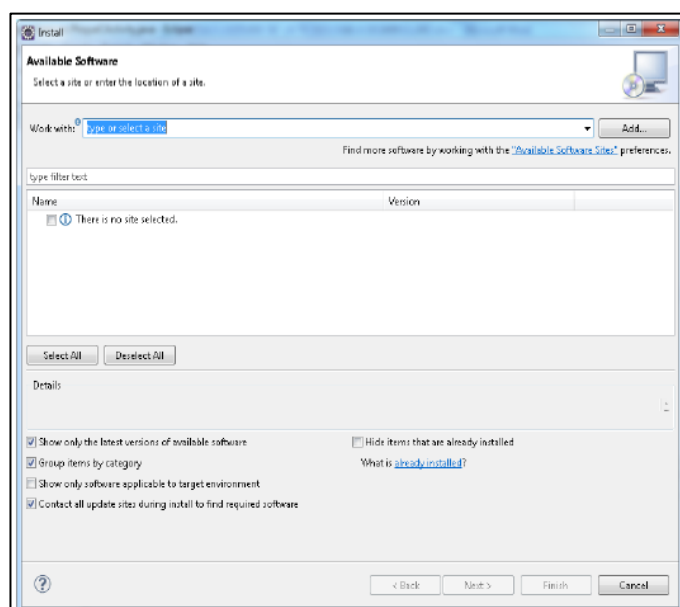


Figura 42

Elaborado por: Edwin Ceballos / Francisco Miño

2. Dar clic en *Arreglar Repositorio*, asignar un nombre al repositorio que se va a agregar y en la localización ingresar el siguiente repositorio web <https://dl-ssl.google.com/android/eclipse/> como se muestra en la figura 43.

Pantalla de asignación de nombre de repositorio que se va a agregar.

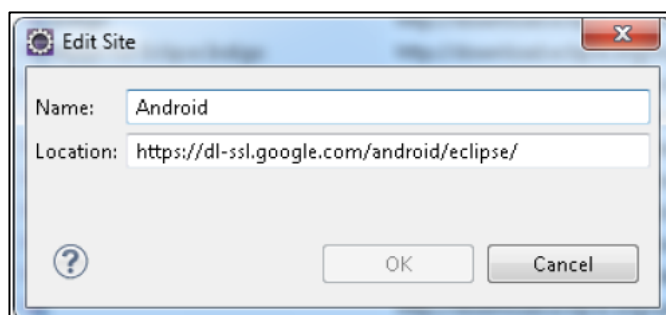


Figura 43

Elaborado por: Edwin Ceballos / Francisco Miño

3. Seleccionar la pestaña *Develoment Tools* y proceder a dar clic en *Siguiente*, lo cual permite instalar los paquetes necesarios para el desarrollo de software de Android, como lo indica la figura 44.

Pantalla de instalación de paquetes para desarrollo de aplicaciones Android.

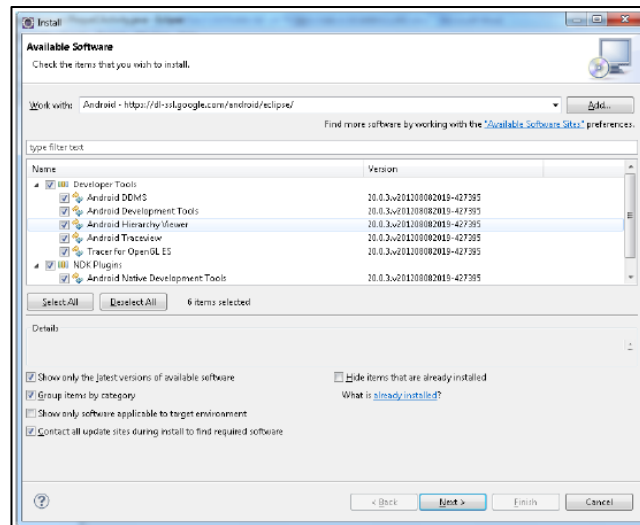


Figura 44

Elaborado por: Edwin Ceballos / Francisco Miño

4. Una vez terminada la instalación se deberá reiniciar Eclipse para que los cambios tengan efecto.

Anexo 7. Construcción del aplicativo EdiBosco Noticias.

Para la construcción del aplicativo EdiBosco Noticias, se debe seguir los siguientes pasos:

1. Seleccionar en el Menú *File > New > Project*, y en el Menú resultante desplegar la opción *Android* y elegir *Android Project* y dar clic en *Siguiente*, como lo indica la figura 45.

Pantalla de nuevo proyecto de Android.

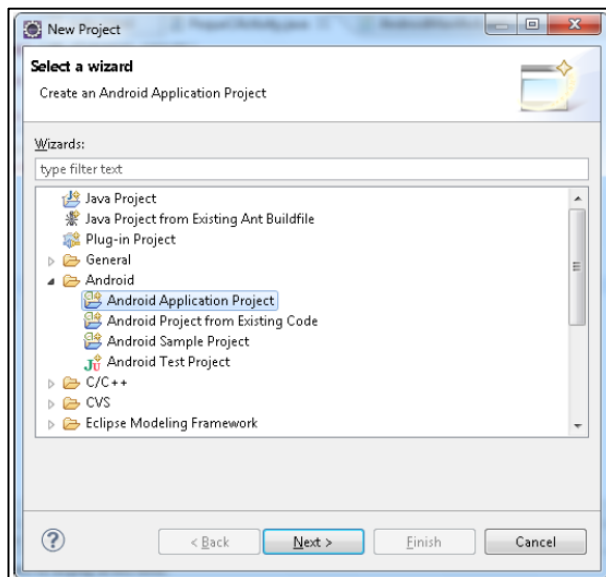


Figura 45

Elaborado por: Edwin Ceballos / Francisco Miño

2. Se presenta la configuración del nuevo proyecto, como lo indica la figura 46, el cual tiene los siguientes parámetros:

- **Project name:** Es el nombre del proyecto, en la práctica será el nombre que reciba la carpeta donde se guardará todo lo relativo al presente proyecto dentro del workspace.
- **Package name:** Es el nombre del paquete bajo el cual será desarrollado todo el código.
- **Activity name:** Es el nombre de la clase Activity principal que será creada, esta clase simplemente es una clase ejecutable, capaz de realizar alguna tarea y es imprescindible en la mayoría de las aplicaciones para Android.
- **Application name:** Es el nombre de la aplicación que se va a desarrollar, constituye el nombre visible para el usuario del dispositivo móvil.

Pantalla de configuración de nuevo proyecto de Android.

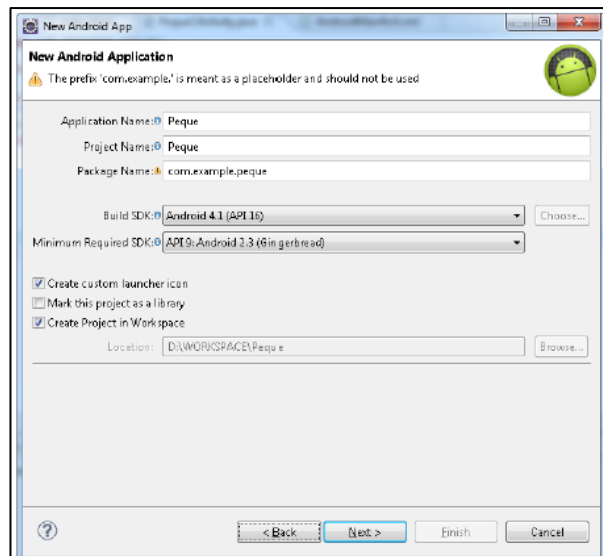


Figura 46

Elaborado por: Edwin Ceballos / Francisco Miño

3. Una vez terminada la creación del nuevo proyecto, se da clic en *Finalizar*.

4. Para ejecutar la aplicación se debe dar clic derecho sobre el proyecto y seleccionar *Run As > AndroidApplication*, en la figura 47 se observa el resultado de ejecutar la aplicación.

Pantalla de ejecución de aplicación móvil EdiBosco Noticias.

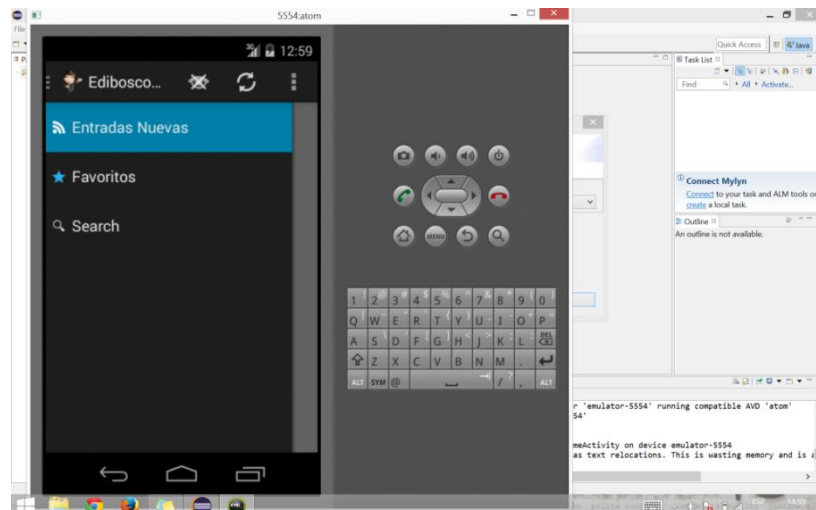


Figura 47

Elaborado por: Edwin Ceballos / Francisco Miño

Anexo 8. Creación de una cuenta desarrollador en Google para publicar la aplicación móvil en la tienda Google Play.

Para la creación de una cuenta desarrollador de Google, se debe seguir los siguientes pasos:

1. Se debe tener creada una cuenta en Gmail, por lo que se procede a crearla, para eso es necesario dirigirse a la siguiente dirección web <http://accounts.google.com/SingUp?service=mail&continue=http%3A%2F%2Fmail.google.com%2Fmail%2F<mpl=default> y se sigue estos pasos:

- Se ingresa los datos personales para la creación de la cuenta.
- Se acepta las condiciones de servicio y las políticas de privacidad de Google y se procede a dar clic en *Siguiente paso*, como lo indica la figura 48.

Pantalla con las condiciones de servicio y políticas de seguridad de Google.

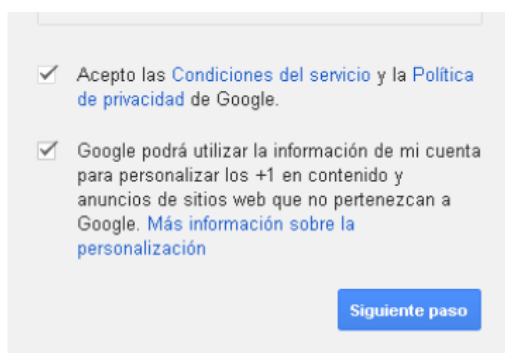


Figura 48

Elaborado por: Edwin Ceballos / Francisco Miño

- Se presenta una pantalla de bienvenida a Gmail, se da clic en Ir a Gmail.
- Finalmente se visualizará la bandeja de entrada de la cuenta creada.

2. Posteriormente se crea un perfil de desarrollador en Google Play, para eso es necesario dirigirse a la siguiente dirección web <http://play.google.com/apps/publish/singup> y se sigue los siguientes pasos:

- Se crea un perfil de desarrollador, para ello se ingresa los datos básicos del desarrollador, luego se da clic en *Seguir*.
- Se lee y acepta el acuerdo de distribución para desarrolladores de Google, se da clic en *Acepto. Continuar*, como lo indica la figura 49.

Pantalla con el acuerdo de distribución para desarrolladores de Google.



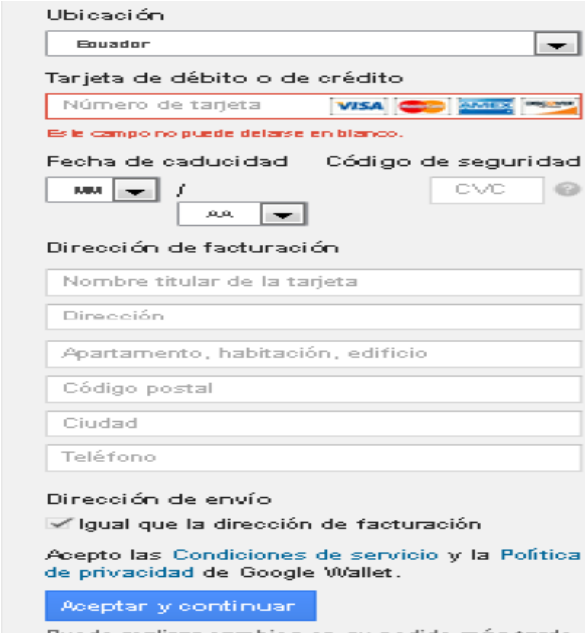
The screenshot shows the Google Play Developer Agreement page. At the top is the Google Play logo. Below it, the title "Acuerdo de distribución para desarrolladores" is displayed. A section titled "Definiciones" lists key terms: Google, Dispositivo, Productos, Market, and Desarrollador o Usted. At the bottom, there is a checkbox labeled "Acepto las condiciones y quiero asociar el registro de cuenta con el Acuerdo de distribución para desarrolladores" which is checked. Below the checkbox is a blue button labeled "Acepto. Continuar »".

Figura 49

Elaborado por: Edwin Ceballos / Francisco Miño

- Por último, se debe pagar una cuota de registro, para ello se especifica un nombre, dirección de facturación y la tarjeta de crédito para el pago de 25.00 dólares, como lo indica la figura 50.

Pantalla de cuota de registro.



The screenshot shows the Google Play registration payment screen. It includes a dropdown menu for "Ubicación" set to "Ecuador". Below is a section for "Tarjeta de débito o de crédito" with a red-bordered input field for the "Número de tarjeta" (marked with a red error message "Este campo no puede dejarse en blanco."), a "Fecha de caducidad" field with month and year dropdowns, and a "Código de seguridad" field. The "Dirección de facturación" section contains input fields for "Nombre titular de la tarjeta", "Dirección", "Apartamento, habitación, edificio", "Código postal", "Ciudad", and "Teléfono". The "Dirección de envío" section has a checked checkbox "Igual que la dirección de facturación". At the bottom, there is a line of text: "Acepto las Condiciones de servicio y la Política de privacidad de Google Wallet.", followed by a blue "Aceptar y continuar" button and a note: "Puede realizar cambios en su pedido más tarde."

Figura 50

Elaborado por: Edwin Ceballos / Francisco Miño

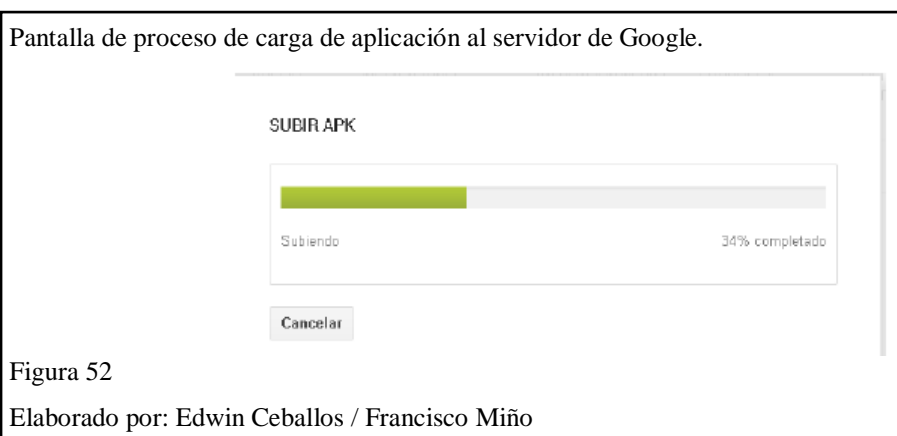
3. Luego de crear la cuenta desarrollador se procederá a publicar la aplicación en Google Play, esta aplicación se la podrá descargar gratuitamente desde cualquier

parte del mundo, para esto es necesario dirigirse a la siguiente dirección web <http://play.google.com/apps/publish>. A continuación se detallan los siguientes pasos:

- Se desplegará el panel de administración de aplicaciones, el cual informa de todas las aplicaciones publicadas. Para subir una nueva aplicación se da clic en *Añadir nueva aplicación*.
- Luego se desplegará una ventana en donde se selecciona el idioma de la publicación y el nombre de la aplicación, luego se da clic en *Subir APK*.
- Se escogerá la aplicación empaquetada que se subirá al servidor, con extensión *.apk*, como lo indica la figura 51.



- La aplicación se empezará a subir al servidor de Google, el tiempo de espera depende del tamaño de la aplicación, como lo indica la figura 52.



- Posteriormente se desplegará una ventana con la información de la aplicación subida, como la versión, dispositivos admitidos y dispositivos excluidos.

- A continuación especificará la información del proyecto, por lo que es necesario dar clic en *Entrada en Play Store*, donde se puede especificar con mayor detalle las características de la aplicación, con los siguientes campos:

Información del producto

- Título.
- Descripción.
- Texto Promocional (opcional).
- Cambios Recientes (opcional).

Recursos gráficos

- Capturas de pantalla.
- Íconos de la aplicación.
- Gráfico de las características (opcional).
- Gráfico Promocional (opcional).
- Video Promocional (opcional).

Categorización

- Tipo de la aplicación.
- Categoría.
- Clasificación de contenido.

Detalles del contacto

- Sitio web.
- Correo electrónico.
- Teléfono (opcional).

Para mayor detalle de cómo especificar el producto hay que dirigirse al sitio web de soporte de Google y posteriormente se da clic en *Guardar*.

4. Finalmente se procederá a especificar el precio de la aplicación y en qué países se distribuirá. Cabe resaltar que esta aplicación será distribuida en todo el mundo de forma gratuita por lo que no se necesita ninguna configuración adicional y se da clic en *Guardar*, como lo indica la figura 53.

Pantalla donde se especifica si la aplicación es gratuita o de pago.

The screenshot shows the 'PRECIOS Y DISTRIBUCIÓN' (Pricing and Distribution) section of the Google Play Store developer interface. At the top, there is a 'Guardar' (Save) button. Below it, the 'Esta aplicación es' (This application is) section has two radio buttons: 'De pago' (Paid) and 'Gratuita' (Free). The 'Gratuita' option is selected. A note below states: 'Para publicar aplicaciones de pago, debes configurar una cuenta de comerciante. [Crea una cuenta de comerciante ahora](#) o [Más información](#).' Below this is the 'DISTRIBUIR EN ESTOS PAÍSES' (Distribute in these countries) section, which indicates that 128 countries have been selected. A scrollable list of countries is shown, each with a checked checkbox. The countries listed are: Albania, Alemania (Germany), Angola, Antigua y Barbuda, Antillas Neerlandesas, Arabia Saudí, Argelia, Argentina, Armenia, and Aruba. A 'Mostrar opciones' (Show options) link is visible next to 'Alemania'.

PRECIOS Y DISTRIBUCIÓN [Guardar](#)

Esta aplicación es ☐ De pago ☒ Gratuita

Para publicar aplicaciones de pago, debes configurar una cuenta de comerciante. [Crea una cuenta de comerciante ahora](#) o [Más información](#).

DISTRIBUIR EN ESTOS PAÍSES

Has seleccionado 128 países y el resto del mundo.

☒ SELECT COUNTRIES

- ☒ Albania
- ☒ Alemania [Mostrar opciones](#)
- ☒ Angola
- ☒ Antigua y Barbuda
- ☒ Antillas Neerlandesas
- ☒ Arabia Saudí
- ☒ Argelia
- ☒ Argentina
- ☒ Armenia
- ☒ Aruba

Figura 53

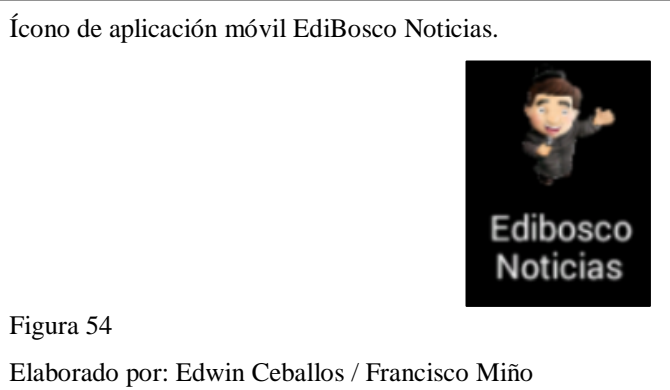
Elaborado por: Edwin Ceballos / Francisco Miño

5. Al concluir se visualizará la aplicación configurada y publicada en la tienda Google Play.

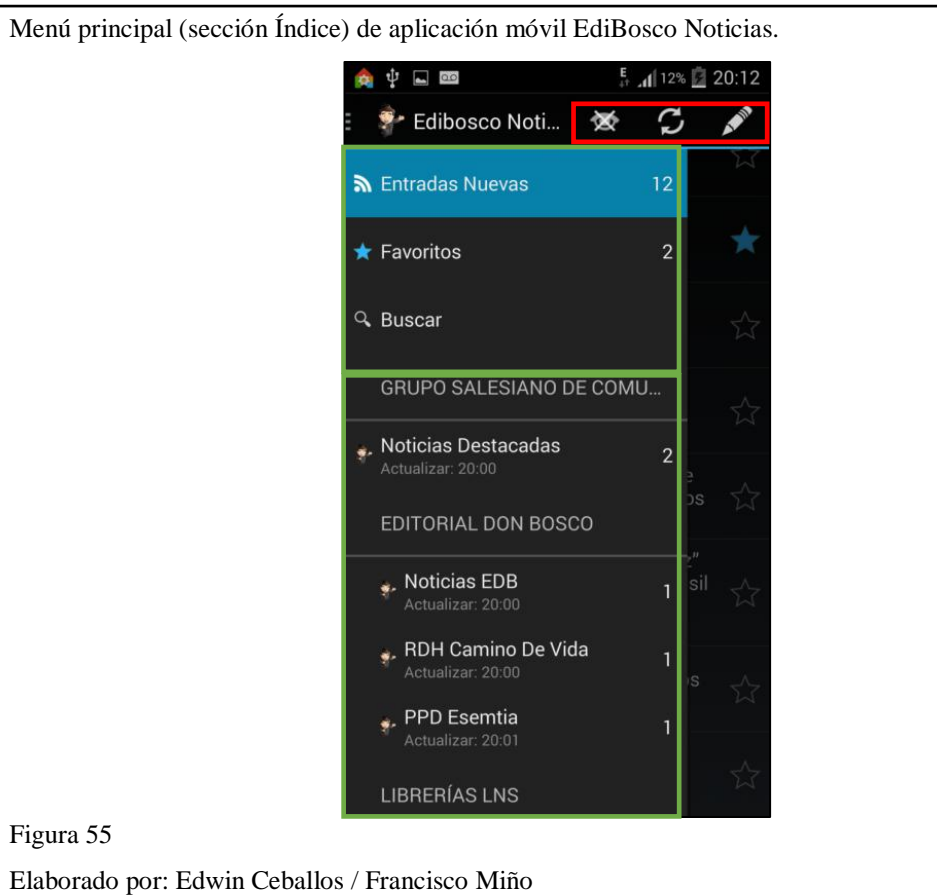
Una característica adicional que brinda Google Play es que se puede visualizar cuadros estadísticos de descargas de la aplicación móvil.

Anexo 9. Manual de usuario.

1. Una vez instalada, buscar en el menú de aplicaciones del dispositivo móvil la aplicación EdiBosco Noticias y hacer un touch sobre el ícono indicado en la figura 54.



2. Se presenta el menú principal (sección Índice), el cual está compuesto por dos grupos de selección de contenidos, los cuales son: Funciones de filtrado y acceso rápido y obras salesianas (representados en los recuadros de color verde), indicados en la figura 55.



3. Al hacer un touch sobre cualquiera de los dos grupos de selección de contenidos, se despliega una lista de noticias, la cual se indica en la figura 56.

Lista de noticias desplegadas en la sección Contenido.



Figura 56

Elaborado por: Edwin Ceballos / Francisco Miño

4. Finalmente, al seleccionar cualquiera de las noticias de la lista con la función touch, se despliega el detalle de la misma, indicado en la figura 57.

Detalle de una noticia seleccionada.

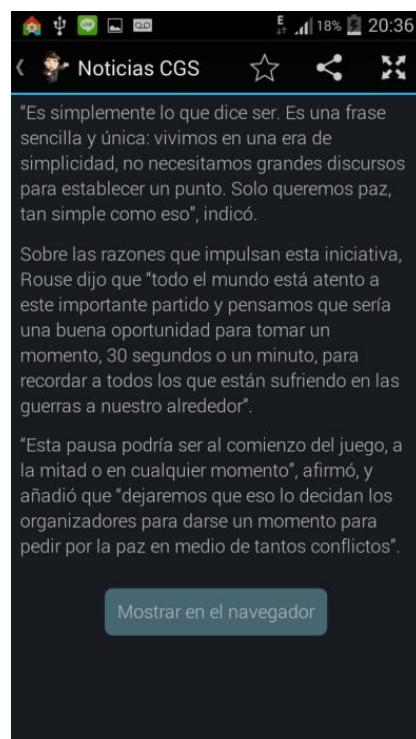


Figura 57

Elaborado por: Edwin Ceballos / Francisco Miño

5. En la pantalla principal de la aplicación móvil EdiBosco Noticias también se encuentra la sección Funciones, indicada en el recuadro de color rojo en la figura 55, que al hacer un touch sobre cualquiera de éstas, se aplican a la información del contenido actual ya sea en las listas o en el panel de lectura de una noticia, algunas son visualizar, compartir, ordenar y refrescar las noticias; y la sección Configuración, indicada en la figura 58, donde se encuentra un listado de opciones administrativas que tienen efecto global sobre la información desplegada en la sección Contenido, al hacer un touch sobre cualquiera de éstas.

Pantalla de sección Configuración de aplicativo móvil EdiBosco Noticias.

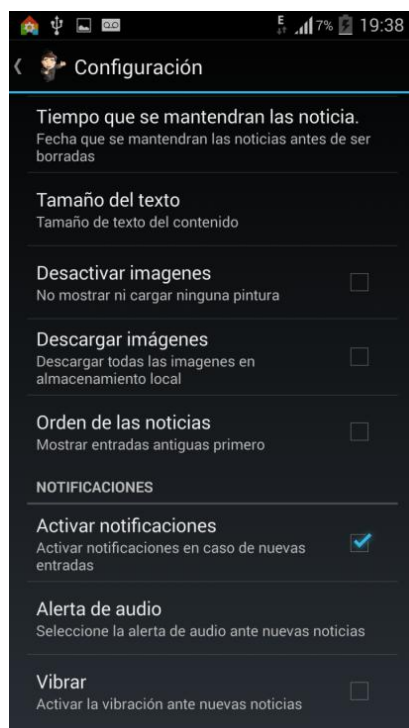


Figura 58

Elaborado por: Edwin Ceballos / Francisco Miño

6. De igual forma se ingresa a la aplicación EdiBosco Noticias al hacer un touch sobre el widget ubicado en el escritorio de Android, indicado en la figura 59, o al seleccionar la aplicación del panel de notificaciones del dispositivo móvil, como se indica en la figura 60.

Widget de aplicación móvil EdiBosco Noticias en el escritorio de Android.



Figura 59

Elaborado por: Edwin Ceballos / Francisco Miño

Ícono de aplicación móvil EdiBosco Noticias en panel de notificaciones de dispositivo móvil.

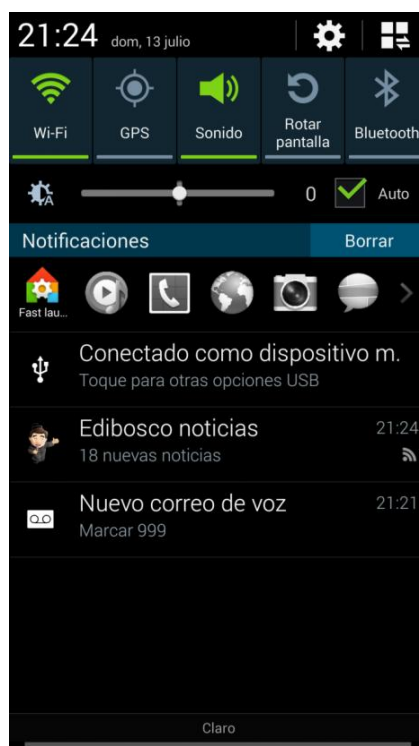


Figura 60

Elaborado por: Edwin Ceballos / Francisco Miño